Computer Science - I - Contents

Distribution of Marks- Questionwise and Topicwise

Sr. No.	Name of Topic	1 Mark Question		3 Mark Question		4 Mark Question		5 Mark Question		Total Mark
		Nos.	Total	Nos.	Total	Nos.	Total	Nos.	Total	
1	Operating system	1	1	3	9	3	12			22
2	Data Structure	1	1	4	12	1	4			17
3	C++	1	1	4	12	2	8	4	20	41
4	HTML	1	1	1	3			2	10	14
	Total	4	4	12	36	6	24	6	30	94

Published By
Mrs. R. Maroo
Kinnari Prakashan, Mumbai - 8

Printed By
Arihant Printers
Marketyard - 37

Chapter 1 OPERATING SYSTEM

Probable marks : 22

Scope of the Syllabus

- What is an Operating System ?
- Services in O.S.
- Overview of Windows 98, Windows NT and LINUX
- Concepts related to Information Management (only definition) :
- File systems, Device drivers and Terminal I/O
- Concepts related to Process Management (only definition) :
- Process, concepts of multiprogramming, context switching, process states, priority, multitasking.
- Concepts related to memory management (only definition) :
- Memory map of single user computer system, partitions, fixed and variable partitions, paging, segmentation and virtual memory.
- G.U.I. (Basic of G.U.I) :
- GUI features such as windows, task list, drag, resize, minimize, maximize, close.

Access and security aspects of O.S. :

- Security threats, attacks on security, computer worms, computer viruses

OPERATING SYSTEM

Q. 1 What is an Operating System ? Write its function.

(Oct. 2004, 2007 Mar. 2009; July 2019, March 2020)

Ans. : An Operating System is a program, which acts as an interface between the user of computer and the computer hardware.

The operating system can be viewed as a set of software programs, normally supplied along with hardware for the effective and easy use of the machine.

The main functions of an operating system are :

i) The primary aim of the operating system is to make the hardware convenient to use.

ii) To help users to execute programs.

iii) To control execution of program to prevent errors and improper use of computer system.

iv) To make provision for security of information to users .

v) To eliminate duplicate errors by number of programmers in development of complicated routines.

vi) It provides facility to share the same hardware among the users.

vii) Proper scheduling of resources among users.

SERVICES IN O.S.

Q. 2 Which are the three main areas in which the operating system divides its services ? Give examples. **(March 2002, 06, 08, 09, 13; Oct. 2004, 07, 14, 15; March 2020)**

Ans. : The O.S. divides its services in the following three main areas :

i) Information Management (IM)

ii) Process Management (PM)

iii) Memory Management (MM)

i) **IM :** Information Management provides facilities to store, retrieve, modify the information on various devices. The services provided under IM are :

(a) create files or directories.

(b) open files or explore directories.

(c) delete, copy or close files.

(d) change working directory.

ii) **PM :** The services provided under process management are directed to keep track of all running programs, called processes. In multiuser operating systems, number of users located at different terminals may execute different programs at a time. In such case operating system keeps track of all processes. It schedules them and dispatches them one after another.

iii) **MM :** The services in Memory Management keep track the of all memory locations. They determine memory allocation policy and use various techniques and algorithms to achieve this.

Q. 3 What is Information Management ? List the system calls in it. **(Oct. 2012)**

Ans. : 1) Information Management provides the facility to store, retrieve, modify or remove the information on files/directories. 2) These system services manage the organization of information into files and directories by allocating memory space to them. 3) It also ensures that correct programs have access to information, have occupied memory space and driving various devices. 4) Some of the system services (system calls) provided under IM are :

i) Create a file.

ii) Create a directory.

iii) Open a file for read/write purposes.

iv) Explore a directory.

v) Close a file.

vi) Read data from file to buffer.

vii) Write data from buffer to file.

viii) Move file pointer.

ix) Create a link.

x) Change working directory.

Q. 4 What is Process Management ? List the system calls in Process Management. **(Oct. 2005)**

Ans. :

1) In multiuser operating system, a number of users, located at different terminals of a network, may execute same or different programs at a time.

2) But such a computer system has only one C.P.U. and it can execute only one instruction, belonging to any one of these programs at the same time.

3) The Process Management of such O. S. keeps track of all running programs called processes, Schedule then and dispatch them one after the another. While doing so, it gives an impression to each user that it has the full control of C.P.U.

4) The Process Management modules of single user O.S. are less complicated than multiuser O.S.

5) The services provided under Process Management are (system calls) :

i) Read a process.
ii) Block a process.
iii) Resume a process.
iv) Terminate a process.
v) Suspend a process.
vi) Delay a process.
vii) Change the priority of a process
viii) Generate a process.

Q. 5 What is Memory Management ? What are the services provided under it ?

Ans. :

1) When a job is to be executed, the O.S. loads the job in main memory from disk.

2) But before loading it in memory, it should know that how much of free memory is available and how much of memory should be allocated to the job.

3) For this, the operating system keeps list of all free memory location. Before a program is loaded in memory the operating system consults with this list. It loads the program into memory and modifies the list.

4) When the program is executed complete, it removes programs from main memory and again modify the list.

5) For this, the O.S. determines memory allocation/deallocation policies and uses various techniques and algorithms to achieve this.

6) The system calls in it are :

i) To allocate a chunk of memory to a process.

ii) To free chunk of memory from a process.

Q. 6 What is system Call ? List any two system calls for memory management. Process management and information management. **(March 2015; July 2017)**

Ans. :

System Call - System calls provides the interface between a process and the operating system. These calls are generally available as n assembly language instruction.

1. System calls in Memory Management - a. To allocate a chunk of memory to a process To free chunk of memory from a process

2. System calls in Process Management - a. Read a process b. Block a process c. Resume a process d . Suspend a process e. Delay a process

3. System calls in Information Management - a. Create a file b. Create a directory c. Open a file d. Close a file e. Create a link f. Move file pointer

Q. 7 What is meant by a system call ? How it is used ? How does an application program (AP) use these calls during execution ? **(Oct. 2003, Mar. 2006)**

Ans. :

1) **System call :** System call provides the interface between a process and the operating system. These calls are generally available as assembly language instructions.

2) System calls are used in different ways as :

(a) Some system may allow system calls to be made directly from a higher level language program. In this, calls are predefined function or subroutine.

(b) Some languages – C, PERL – allow system calls to be made directly.

(c) FORTRAN system provides set of library routines.

3) An application program uses sequence of system calls during execution. To prompt a message on the terminal AP uses system call. Next, read from the terminal AP uses another system call. Similarly for each and every task, program uses various system calls.

OVERVIEW OF OPERATING SYSTEM

Q. 8 What are the features of Windows 98 ? **(Oct.2004, 13, March 2015, 2020, July 17)**

Ans. : The main features of Windows 98 are as listed below :

i) Easier to use :

1) Windows 98 is a single user multitasking operating system.
2) Navigating around the computer is easier in Windows 98.
3) A file can be opened by a single click.
4) Windows 98 allows us to use multiple monitors with single computer.
5) New hardware can be easily installed and used without restarting computer.
6) With Windows 98 we can use digital cameras and other digital imaging devices.

ii) Faster :

1) Windows and programs open faster than in Windows 95.
2) The computer speed and efficiency can be easily improved by simple maintenance.

iii) True web integration :

1) Windows 98 can be easily connected to internet.
2) Web pages can be viewed in any window.
3) Using Microsoft Outlook Express, E-mails and message can be send to internet newsgroups.
4) The internet conferences can also be arranged.

iv) More entertaining :

1) Windows 98 supports DVD and digital audio. User can play high quality digital movies and audio on the computer.
2) The television broadcast can also be seen.

). 9 What are the features of Windows NT? OR **(Mar.2011, 2016, 2017)**

Explain any four features of Windows NT operating system. **(Oct.2002, 2006, 2012, 2013, July 2017)**

ıs. : The features of Windows NT are as listed below :

i) Windows NT is multitasking, multiuser and multithreading operating system.

ii) A user will get faster response eventhough multiple applications are running.

iii) Windows NT supports virtual memory management system to allow multiprogramming.

iv) Symmetric multiprocessing in windows NT allows it to schedule various tasks on any C.P.U. in a multiprocessor system.

v) Windows NT is a 32-bit operating system.

vi) Windows NT uses New Technology File Systems (NTFS), which implements fault tolerance, security and has support for very large files.

Q. 10 What are the features of Linux ? **(March 2004,2014, 2019, July 2017,.2018) (Oct 2013, 2014)**

Ans. : Some of the features of Linux are as given below :

i) Linux is a multiuser, operating system with a full set of unix compatible tools.

ii) Linux runs on a wide variety of platforms. It was developed exclusively on PC architecture.

iii) It provides as much as functionality from limited resources. It can run on machine having 4MB of RAM.

iv) Linux presents standard interfaces to both the programmer and user.

v) Linux supports a wide base of applications.

vi) Linux is free software. Free in the sense that people can copy it, modify it, use it in any manner they want.

vii) The file system in LINUX obeys UNIX semantics.

Q. 11 What are the components of Linux system ? **(March 2004)**

Ans. :

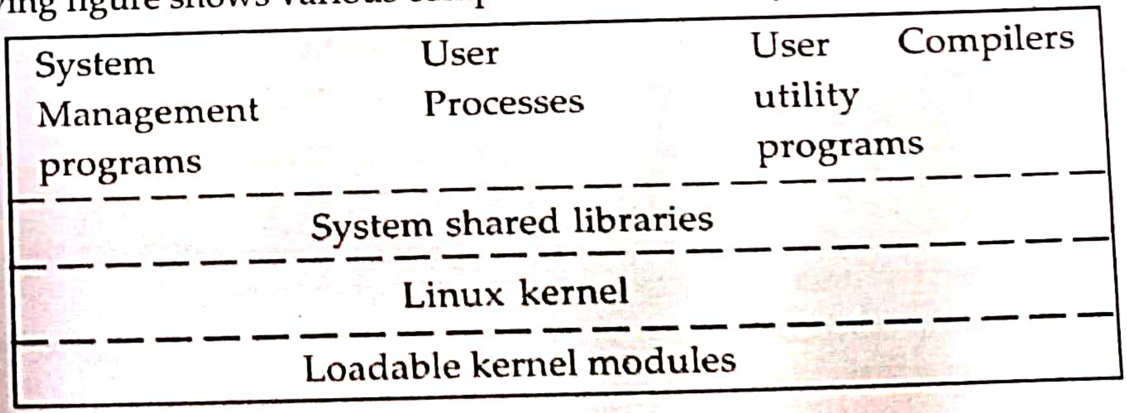
The Linux is composed of three main bodies :

1. **Kernel :** Kernel maintains all important abstractions of the operating system, such as processes and virtual memory.

2. **System libraries :** System libraries define a standard set of functions through which applications can interact with the kernel, and which implements much of the O.S. functionality.

3. **System utilities :** These are programs that perform individual, specialised management tasks. Some system utilities may be invoked just once to initialize and configure some aspects of system.

Following figure shows various components of Linux system :



CONCEPTS RELATED TO INFORMATION MANAGEMENT

Q. 12 What is a file system ?

Ans. : **(March 2015; 2018; July 2018)**

1) The collection of related information i.e. data or programs is called as file.
2) Each file has a specific name, which is used to refer that file.
3) For convenient use of the computer system, the O.S. provides a uniform logical view of information storage.
4) The operating system manages mass storage devices to implement the abstract concept of file.
5) The O.S. maps files on to physical devices such as tapes or disks.
6) Using various data structures, file system in IM allows user to define files and directories and allocate/deallocate the disk space to each file.
7) There are two types of file systems :
   i) Tape - based systems ii) Disk - based systems.

i) **Tape-based systems :**
   (a) Tape-based systems are simple but inefficient.
   (b) In these systems, files are stored on to reels of physical tapes. Generally one or more files are stored on to one tape.
   (c) Tapes are used for transport of data from one computer to another.

ii) **Disk - based systems :**
   (a) Each disk is divided into tracks and each track is further divided into number of sectors.
   (b) Number of tracks and size of sectors is variable. It varies from one drive to another.
   (c) A disk has a device directory, indicating, which files are on the disk. The directory lists the file name, starting address, file length, type of file, time of creation, and time of last update etc.

Q. 13 What a function of file system and device management system?

Ans.: Function of file system are:

1. It allows the user to define files and directories and allocated/deallocate the disk space to each file.
2. It stores data or information.
3. Arrange various file of same type under one directory.
4. Open a file; create a file, delete a directory and set certain access controls on a file.

Function of device management or device driver are :

1. Written special subroutine for each I/O device such subroutine is called device driver.
2. It moves data from driver to terminal.
3. Block numbers are converted into sector number.

Q. 14 What are the advantages of disk-based systems over tape-based systems ?

Ans. :

Advantages of disk-based systems over tape - based systems are :

i) Finding a file on tape-based systems is difficult and time consuming, while a file can easily be found on a disk-based system.

ii) In tape-based systems, if a file is to be modified, it requires to copy entire tape whereas in a disk based systems a file can be modified easily, without copying the entire disk.

iii) Store large information in disk-based systems as compared to tape-based systems.

iv) Disk-based systems are easier and convenient to use instead of tape-based systems.

v) Each disk consists of number of blocks, which can be rewritten easily, while we require entire tape to copy, if we have to rewrite something.

Q. 15 Explain three stages of I/O operations related to disk?

OR Explain why user is not allowed to directly interact with the hard disk. **(March 16)**

Ans.:

I/O operations required to read data from disk. These operations are as follows :

To move R/W heads from the current address to the target address three stages are :

1. The time taken to move R/W head in or out to position on the correct track is called seek time.
2. The time taken to wait until the desired sector comes under R/W head as the disk rotates is called latency or rotational delay.
3. The time taken to activate R/W head for appropriate surface and read data is known as transmission data.

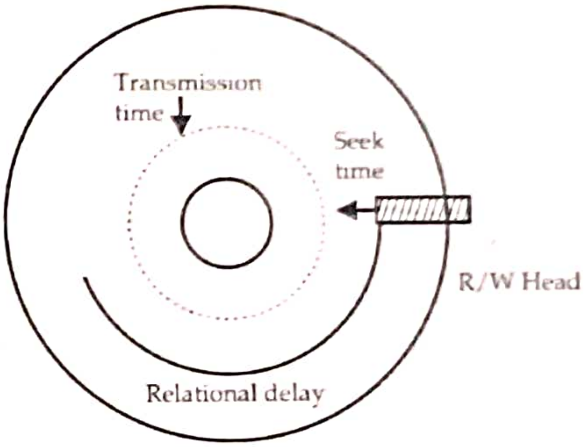


Fig. Q. 15

Q. 16 Explain the file system related to Information Management with file operations only. **(March 2004, 2007; Oct. 2015)**

Ans. :

1) The file system related to IM allows the user to define files and directories and allocate/deallocate the disk space to each file.

2) A file is a collection of related information. It can be program or data.

3) **File operations :** The file concept is implemented by the operating system. System calls are provided to create, read, write, rewind and delete files.

   (a) **Create a file :** For creating a file, first whether sufficient space is available for that file is checked. If it is available, entry for new file must be made in directory.

   (b) **Write to a file :** For writing to a file, there is command in which name of the file is given. Then operating system search for that file in directory entry and write to it.

(c) **Reading a file :** For reading a file, there is a system call in which file name is specified. Then operating system searches for that file in directory entry and read it.

(d) **Rewind a file :** The directory is searched for appropriate entry and file is reset to the beginning of file.

(e) **Delete a file :** To delete a file, directory entry is searched and if file is found, it releases the memory space and that directory entry now become invalid.

Q. 17 Define a block with reference to Operating System. What are the parameters of a block of data that are concerned with an Operating System. Explain in brief. **(March 2012)**

Ans.: Definition of block :

A block is a logical unit of data that operating system defines for its convenience. Thus block is a contiguous set of bits or bytes that forms an identifiable unit of data.

Operating system has following parameters of a block

1. File ID
2. The starting position in the file
3. The number of bytes to be 'read
4. The starting address of memory Where data. is to be read.

1. File ID – It is a letter code given to each type of data file to make it easier for the operating system to identify the files.
2. The starting position in the file – It is the address of memory location from where file will start.
3. The number of bytes to be read – It is total size of file in bytes.
4. The starting address of memory – It is starting address of block.

Whenever program read any data the file system translates request into reading one or more sector from disk and instruct device driver to read these sector O.S. keeps all its data structure in terms of block O.S. translates a block number into sector number. File system request to read desired blocks. It uses disk space allocation i.e. linked list to carry out translation. File system then request device driver to read desired block. DD issues instruction to the controller for the disk to read required blocks. Controller reads data sector by sector and stores it in its own memory until desired block are read in.

Q. 18 Define a Block of data. State necessary parameters to be provided to O.S. to access a block of data. **(March 2014)**

Ans. :A block is a physical unit of data on the disk. Block consist of one or more contiguous sector. Parameters to be provided to O.S. to access block of data

1. **Sector Number (SN)** - Entire disk view as a series of sector from 0 to n as sector number.

2. **Relative Byte Number (RBN)** - O.S. calculate RBN for each record. This is starting byte no. of each record.

Relative Record Number (RRN) - O.S maintains a field called Cursor to keep track of current RRN which is incremented after each record is read / written.

Physical Block Number (PBN) - O.S. translate logical block number into Physical Block Number.

Physical Sector Address - While reading a record, if the physical address is maintained in the index. DMS itself can access the final address without going to go through various level of address translation.

6. **Logical Block Number (LBN)** - File system of OS calculate the Logical Block Number as integer value of RBN /512.

7. **Logical Address** - File System of OS translate logical address into Physical address.

8. **FAT (File Allocation Table)** - Before any file is written or read, OS brings the blocks containing FAT entries in memory for future operation.

Q. 19 Explain the following terms in case of magnetic disk :

(i) Tracks and Sectors (ii) Seek time

(iii) Transmission time (ii) Latency time/ Rotational delay

(Mar. 2006, 2009 Oct. 2003, 2006, July 2017)

Ans. :

i) **Tracks and Sectors :** Magnetic disk surface is made up of concentric circles called tracks. The number of tracks varies depending on the disk type. A track is further divided into smaller areas called sectors.

A sector is a smallest unit of information which can be read from or written to the disk. Sector varies from 32 bytes to 4096 bytes and track contains 4 to 32 sectors per track and from 75 to 500 tracks per disk surface.

ii) **Seek time :** The time required for read/write heads to move to the correct track is called as seek time.

iii) **Transmission time :** The time required for activate Read/Write head for appropriate surface and read data is called as transmission time.

iv) **Latency time/Rotational delay :** The time required for requested sector on track to rotate below the head is called as latency time or rotational delay.

Q. 20 Explain internal and external fragmentation. **(March 2020)**

Ans. : Internal fragmentation :

Wastage of memory space within partition is called as internal fragmentation. A file consists of number of blocks. Consider the block size of a O. S is 1024 bytes and a file is of 3499 bytes.

Then, when it is loaded in memory for execution it would have allocated 4 blocks. Thus last 597 bytes would be wasted. This is called as internal fragmentation.

Larger block size causes more internal fragmentation.

External fragmentation :

Variable partition suffers from external fragmentation.

Suppose a job of 512 bytes is terminated and new job is of 256 bytes is loaded in the partition, then 256 bytes of memory is wasted. This is called as external fragmentation.

Q. 21 What are device drivers? **(March 2010; July 2018)**

Ans. : 1) Device drivers are software programs required for each device.

2) Each device will require different drivers as per functionality.

3) A device driver knows how the buffers, flags, register control and status bits should be used for a particular device.

4) Some device drivers are useful for data conversion.

5) For simply reading a character from a device involves complex sequence of device specification operations.

6) Rather than writing the code every time, the device driver was simply used from library.

Q. 22 What is VDU ? Explain the following terms :
a) Dumb terminal b) Intelligent terminal

(March 2002, Oct. 2007, March 2018; July 2019)

Ans. :

1) A visual display unit (VDU) or terminal is nothing but a common I/O medium.

2) There are generally two parts of terminal I/O hardware :
   i) Input unit e.g. keyboard
   ii) Output unit e.g. screen

3) There are two types of terminals :
   i) Dumb terminal
   ii) Intelligent terminal.

i) Dumb terminal :

(a) A dumb terminal consists of a microprocessor on which small programs can run and limited amount of memory.

(b) It is responsible for basic I/O operations.

(c) A dumb terminal does no processing on input characters.

ii) Intelligent terminal :

(a) An intelligent terminal has powerful hardware and software than a dumb terminal.

(b) It also has more amount of memory.

(c) It can carry out some process on input characters also it can process programs independently.

Q. 23 Explain the use of video RAM. Explain data bytes and attribute bytes.

(Mar. 03, 08, 11, 13; Oct. 13, July 16)

Ans. :

i) The video RAM is basically the memory that the terminal hardware itself has.

ii) Anytime all characters stored in the video RAM are displayed on the screen by the video controller using display electronics.

iii) All particular informations (attributes) are stored in video RAM.

iv) Video RAM consists of 2000 databytes (0 to 1999) preceded by 2000 corresponding attribute bytes (0 to 1999) as shown in figure.

Attribute Byte 0	**Data Byte 0**	**Attribute Byte 1**	**Data Byte 1**		
				Attribute Byte 1999	Data Byte 1999

v) A typical alphanumeric screen can display 25 lines each consisting of 80 characters i.e. 25 × 80 = 2000 characters.

vi) **Data Byte :** All 2000 characters are stored in video RAM. To display any specific character on the screen at a specific position all ASCII or EBCDIC code for that character is to move in video RAM.

vii) **Attribute Byte :** There is one attribute byte for each data byte. This byte tells the video controller how the character is to be displayed. It signifies whether the corresponding data character which is stored next to it in the video RAM is to be displayed bold, underlined, blinking or in reverse video etc.

Q. 24 Explain in brief : (ii) Types of Memory Mapped I/O Terminals. **(March 2010)**

Ans. :

Types of Memory Mapped I/O Terminals:

Two types of Memory Mapped I/O Terminals;

i) Alphanumeric (Character oriented) : **Please refer ch1/Q-23/p-1.-10)**

ii) Graphics (Bit oriented) : 1) For bit oriented color graphics terminals require 24 or 32 for each byte or bit. 2) This increases Video RAM capacity requirement.

Q. 25 Why keyboard is referred as memory map terminal? Explain the multiple memory location involved in the input-output operations between the keyboard and screen. **(March 2005, 2010, Oct.2010, July 2016)**

Ans. :

(a) Terminals have a video RAM generally with 2000 data bytes preceded by 2000 corresponding attribute bytes.

(b) Anytime, all the 2000 characters (25 lines × 80 columns on screen) stored in video RAM are displayed on the screen by the video controller.

(c) The video RAM is treated as part of the main memory only.

(d) Therefore, for moving any data in or out of the video RAM, ordinary load or store instructions are sufficient.

(e) So, keyboard is referred as memory map terminal.

The following multiple memory locations are involved in the input-output operations between the keyboard and screen :

(a) **Small memory within the keyboard itself :** When a character is keyed in, the 8-bit ASCII EBCDIC code is generated which is stored temporarily in the memory of the terminal itself.

(b) **The video-RAM (data and attribute byte) :** The ASCII or EBCDIC code for the character is to move to the video RAM at the corresponding position with appropriate co-ordinates.

(c) **The operating system buffers :** The operating system has one buffer for each terminal and two separate buffers for input and output operations.

(d) **The I/O area of the application program :** When the user finishes keying in the data, the data stored in operating system buffer for that terminal is flushed out to the I/O area of the application program which wants that data.

CONCEPT RELATED TO PROCESS MANAGEMENT

Q. 26 What is a process ? **(Oct. 2002, Oct. 200**

Ans. : *Process :*

i) A process is defined as a program under execution, which competes for CPU time and other resources.

ii) In simple terms, a program does not compete for computing resources such as C.P.U time or memory, whereas a process does. A program may be present on paper or reside on disk. It may be compiled or tested but it still does not compete for computing resources.

iii) Once a user wants to execute a program, it is located on the disk and loaded in the main memory, at that time, it becomes a process, because it then compete for C.P.U. time and other resources.

Q. 27 Define the terms : 1) Context switching 2) Degree of multiprogramming.

Ans. :

1) Context switching :

In multiprogramming system, multiple processes are run at the same time such that when process 1 wait for an I/O, process 2 executes and vice versa. The lost in time, in turning the attention of CPU from one process to another is called as context switching.

2) Degree of multiprogramming :

The number of processes running simultaneously and hence competing for CPU is known as degree of multiprogramming.

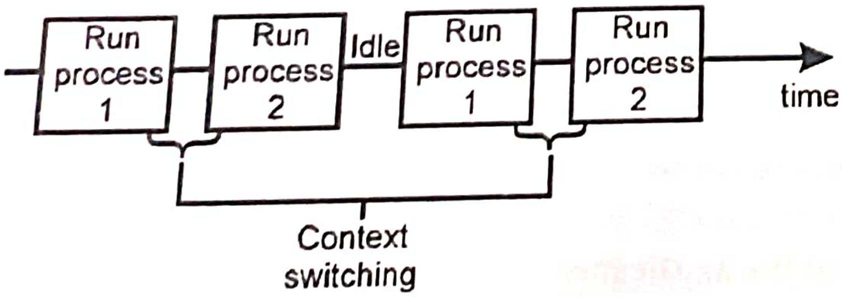
Q. 28 Explain context switching at a process level in multiprogramming system with example. **(March 2002, 2009, 2011, 2016; Oct. 2010, 2015, July 2016)**

Ans. :

i) Multiprogramming is the concept of increasing utilisation of C.P.U. by always having something for C.P.U. to execute.

ii) In multiprogramming, C.P.U. can execute two or more processes simultaneously. When process 1 waits for an external event such as an I/O operation, C.P.U. executes process 2 and vice versa.

iii) When C.P.U. switches from one process to another the time required for switching is called as context switching.



iv) Let A and B be the two processes ready for execution and requires C.P.U. time for execution.

Let CPU time be given to process A, which is having some instructions depending on process B or on some external event such as an I/O operation. Then, it is the job of operating system to halt the execution of process A and give C.P.U. time for process B.

The lose in time in turning the attention of C.P.U. from process A to process B is known as context switching.

v) During context switching the status of C.P.U. registers and flags of the old process are stored in memory.

vi) For-e-g :-

```
          // A. CPP
          # include <iostream.h>
          # include "B.h"
          void main ()
          {
          int a = 10, b = 20;
          cout << add (a,b);
          }
          // B.h
          # include <iostream.h>
      int add (int x, int y)
          {
          return (x+y);
          }
          :
          .
```

Here A and B are the two processes, where process A is depending on B. When process A is being executed, CPU executes instructions one by one. When it comes to the instruction cout <<add (a,b); it stops execution because the output of this instruction depends on Output of process B.

Thus there is need of execution of process B. So C.P.U. stores the contents of registers and flags of process A in RSA (Register Save Area). It then loads process B in memory. The time required for this is known as context switch. The CPU then executes process B and output is given to process A and execution of process A restarts from the instruction from which it was halted.

Q. 29 Explain Running, Ready and Blocked process states in process management. OR Discuss various process states with examples.

(Oct. 2002,04,05,12 Mar. 2012, 13, 14, March 2018; July 2019)

Ans. :

In order to manage switching between processes, the operating system defines three basic process states, which are as given below.

i) **Running state :**

There is only one process, which is executed by C.P.U. at any given moment. This process is called as running process. In multiprocessor systems, with multiple C.P.Us, there are many running processes at a given moment. The operating system keeps track of all of them.

Ready state :

The process, which is not waiting for an external event such as an I/O operation, b which is not running is said to be in ready state. Actually, a process in ready state co have been running. But the fact that, there is only one C.P.U., which is executing so other process, while this process is waiting for C.P.Us attention towards it.

iii) **Blocked state :**

When a process is waiting for an external event such as an I/O operation, the process said to be in blocked state. The major difference between blocked and ready process that a blocked process can not be directly scheduled even if CPU is free, whereas ready process can be scheduled if the C.P.U. is free.

Q. 30 What is process scheduling ? Explain scheduling objectives. **(July 201**

Ans. :

1) In a Multiuser Operating System, a number of programs are running simultaneously and these are called as processes.

2) In this case the O.S. has to keep track of all these processes and will have to dispatch them one after another. This is known as process schedulin.

3) While scheduling various processes, there are many objectives for the operating system. Some of these objectives conflict with each other. Therefore, the O.S. designers have to choose a set of objectives to be achieved. Some of these objectives are as listed below :

i)	Fairness	ii)	Good throughput
iii)	Good CPU utilization	iv)	Low turnaround time
v)	Low waiting time	vi)	Good response time.

i) Fairness refers to being fair to every user in terms of C.P.U. time

ii) **Throughput** refers to the total productive work done by all the users put together.

iii) CPU utilization is the fraction of the time that the CPU is busy.

iv) **Turnaround time** is the elapsed time between the time a program or job is submitted and the time when it is completed.

v) **Waiting time** is the time a job spends waiting in the queue of the newly admitted processes for the operating system to allocate resources to it before commencing it execution.

vi) **Response time** is the time slice for responding to a question or an event. It depends o degree of multiprogramming, the efficiency of the hardware along with the O.S. and t policy of O.S. to allocate the resources.

Q. 31 Explain the following terms in case of process scheduling : **(Mar. 2005) (Oct. 2008, March 201 (July 201**

(a) Turnaround time (b) Waiting time

(b) Terminal response time (d) Event response time

Ans. :

(1) **Turnaround time :** Turnaround time is the elapsed time between the time a program a job is submitted and the time when it is completed.

2) **Waiting time :** Waiting time is the time a job spends waiting in the queue of the new admitted processes for the operating system to allocate resources to it bef commencing its execution.

Terminal response time : In the Time-sharing system, Terminal response time is the time to respond with an answer or result to a question and it depends on degree of multiprogramming, the efficiency of hardware with OS and policy of OS to allocate resources.

Event response time : In the real-time system, event response time is the time to respond with an event.

2 What are preemptive and non-preemptive philosophies of scheduling.

. : There are basically two scheduling philosophies, depending upon the need. The rating system designers has to select one of them.

emptive philosophy :

This philosophy allows a higher priority process to replace a currently running process, even if its time slice is not over or it has not requested for any external operation.

This requires context switching more frequently.

It is suited for on-line, real time processing, where interactive users and high priority processes require immediate attention.

The preemptive philosophy increases fairness of the system but decreases throughput.

n-preemptive philosophy :

In a non-preemptive philosophy, a running process retains the control of the CPU and all the allocated resources, until it surrenders control to the operating system (on its own wish).

This means that even a high priority process enters the system, the running process can not be forced to give up control.

) However if the running process is blocked due to some external request another process can be scheduled.

) It is not suited for real time systems, where high priority events requires an immediate attention.

) Non-preemptive philosophy increases throughput of the system but decreases fairness.

). 33 What is mean by preemptive scheduling? How does it affect the system performance verses non-preemptive scheduling?

ns.: Preemptive scheduling allows a higher priority process to replace a currently running process, even if its time slice is not over or it has not requested for any I/O. This requires context switching more frequently. It is used for on-line, real time processing.

At railways reservation system or a bank is concerned with bookings, cancellation and nany types of enquiries response time is very crucial otherwise customer satisfaction will be oor. In such a case preemptive scheduling is better.

In case of non-preemptive scheduling running process cannot be forced to give up to control.

Q. 34 What is priority ? Explain internal and external priorities. **(Oct. 2007)**

Ans. :

1) The concept of arranging ready processes in a queue so that they can be dispatched one after another for execution depending on some policy is known as priority.

2) Due to many processes competing for the same available resources like C.P.U. and memory, concept of priority is used.

3) A priority may be external (or global) or internal (or local).

(a) External or global priority :

(1) An external priority is specified by the user externally generally at the time of initiating the process.

(2) In many cases, the operating system allows user to change its priority externally even during its execution.

(3) If the user does not specify any external priority at all, the operating system assumes a certain priority, called the default priority. But when an urgent job needs to be done, the system manager permits the process to be created with a higher priority.

(b) Internal priority or local priority :

(1) The concept of internal priority is used by scheduling algorithms. They base their calculations on the current state of the process e.g. each user, while firing a process, can be forced to specify the expected time that the process is likely to take for completion.

(2) The operating system can then set internal priority, which is highest for the shortest job (SJF i.e. shortest job first algorithm), so that at only a little extra cost to large job, many short jobs will complete.

(3) This has two advantages : (a) If short jobs are finished faster, the number of processes competing for C.P.U. will be decreased. (b) The number of satisfied users will increase.

(4) However if a stream of short jobs keeps coming on, an important large job may suffer from indefinite postponement. To avoid this, set higher external priority to important large jobs.

Q. 35 With reference to process management explain the terms:

(i) External priority **(ii) Purchased priority**

(iii) Internal priority **(iv) Time slice** **(Mar. 2003, 06, 08, 11,16,17, 19)**

Ans. :

(i) External Priority : Please refer Q. No. 34.

(ii) Purchase priority :

(a) This priority is used in some data centre situations where each user pays for the time used.

(b) Higher priority processes are charged at a higher rate to prevent each user from firing his job at the highest priority. This is known as scheme of purchased priority.

(c) Operating system keeps track of the time used by each process and the priority at which it was used.

(iii) Internal priority : Please refer Q. No. 34.

(iv) Time slice :

(a) Each process is normally given certain time to run irrespective of its importance. This is known as Time slice.

(b) Time slice given to each process so that a process does not use the CPU indefinitely

Q. 36 What is multitasking? Explain in brief.

Ans. :

1) A task can be defined as an asynchronous code path within a process.

2) Hence in operating systems which supports multitasking, a process can be considered to be made up of number of tasks, which can run simultaneously in the same way that a multiuser operating system supports multiple process at the same time.

3) Just like processes, a task can also have priorities and states.

4) A task can be in ready, running or blocked states and accordingly task control blocks (TCB) are linked together.

5) When the operating system schedules a process with multiple tasks and allocates time slice to it, the following happens :

   (i) The operating system select the highest priority ready task within that process and schedules it.

   (ii) At any time if the process time slice is over, the operating system turns the process as well as currently running task into ready state from running state.

   (iii) If the process time slice is not over but the current task is either over or blocked, the operating system chooses next highest priority ready task within that process and schedules it.

   (iv) If there is no other ready task within that process only then the O.S. turns that process to blocked state.

   The multitasking operating system provides "Inter Task Communication" & "Task Synchronization" for communication between different tasks.

Q. 37 Explain the term multitasking with a suitable example. **(Oct. 2003, 2013)**

Ans. :

1) **Multitasking :** A task can be defined as an asynchronous code path within a process. In multitasking, a process can consist of tasks, which run simultaneously.

2) Multiple tasks should be able to run concurrently within a process.

3) Multitasking allows programmer flexibility and improves CPU utilization.

4) It reduces the overheads of switching at a process level.

5) For example :

   Consider a process consisting of two tasks :

```
    Task 0 :     Read a Record
                     Process a Record

              Task 0 end
    Task 1 :     Write a Record
              Task 1 end
```

   (a) Two task are defined within the same process. They run concurrently within the same process if synchronized properly.

   (b) If task 0 is blocked, instead of blocking entire process, the operating system will find out whether tasks 1 can be scheduled.

(c) When both tasks are blocked, only then entire process is blocked. Again if one task is ready, the process can be moved to ready list and then scheduled.

Q. 38 What are the advantages of multitasking operating systems ?

Ans. :

(i) Multitasking O.S. allows programmer flexibility and also improves C.P.U. utilization.

(ii) When various tasks are defined in a process, then process would be blocked only if all the tasks in that process are blocked.

(iii) Again even if any task becomes ready, the process can be moved to ready list from blocked list.

(iv) By adding task levels, context switching at various process levels can be reduced.

(v) Multitasking is less time consuming and it reduces turnaround time.

Q. 39 Explain multiuser and time sharing operating systems. **(March 2015, 2020)**

Ans. : In computing, **time-sharing** is the sharing of a computing resource among many users by means of multiprogramming and multi-tasking.

Time sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing. The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, objective is to maximize processor use, whereas in Time-Sharing Systems objective is to minimize response time.

Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receives an immediate response. For example, in a transaction processing, processor execute each user program in a short burst or quantum of computation. That is if n users are present, each user can get time quantum. When the user submits the command, the response time is in few seconds at most.

Operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are following

(i) Provide advantage of quick response.

(ii) Avoids duplication of software.

(iii) Reduces CPU idle time.

Disadvantages of Timesharing operating systems are following.

(i) Problem of reliability.

(ii) Question of security and integrity of user programs and data.

(iii) Problem of data communication.

Q. 40 Explain various disk space allocation with their merits/demerits?

Ans.: There are two major types for the allocation of disk space to files these are:

1. Contiguous allocation 2. Non-contiguous allocation

1. **Contiguous allocation :** It requires each file to occupy a set of contiguous address on the disk. An unallocated segment is called hole. If a new file is created and there are set of holes then following strategies is used. Select a free hole from set of holes

   1. First Fit - This allocate first hole that is big enough.
   2. Best Fit - This allocate the smallest hole that is big enough.
   3. Worst Fit - This allocates largest hole.

   Mertis - If processing is sequential and if the operating system uses buffered I/O, the processing speed can increase.

   Demerits - Disadvantages of contiguous allocation is space wastage and inflexibility.

2. **Non-contiguous allocation :** In this maximum size of the file does not have to be predicted at the beginning. The file can grows with time as per needs. This reduces wastage of space.

   Mertis - Operating system automatically allocates additional blocks, if the file gets full during the execution of program without aborting the program.

   There are two methods of non-contiguous allocation these are.

   1. Chained allocation 2. Indexed allocation

1. **Chained allocation :** It is used in non-contiguous allocation. In this each file is a linked list of disk blocks, the disk blocks may be scattered any where on the disk. The directory contains a pointer to the first and last block of the file. Create a file in this allocation is easy.

   Merits - 1. There is no external fragmentation 2. There is no need to declare size of file when it is created.

   Demerits - 1. Each file require space for pointer 2. If a pointer is lost then can't open file.

2. **Indexed allocation :** In this all pointer are brought together into one location called indexed block. Each block has its own index block, which is an array of disk block address. An index can be a list of pointers. Index allocation support access, without suffering, from external fragmentation. The index block is normally on the disk block which can be read or written by itself.

Q. 41 Contrast contiguous verses non-contiguous memory management system.

Sr. No.	Contiguous memory management	Non-contiguous memory management
1.	This system expect the program to be loaded in contiguous memory location.	in this system program is divided into different chunks and to be loaded at different portions of the memory.
2.	Example - Single contiguous memory management system, Fixed partition memory management system.	Example - Paged memory management system, segmented and combined system, segmented and combined memory management system.

...ctions are performed by memory management of operating systems ? S... any four memory management systems.

(March 2003,04,07,12,13,14, 17,19; Oct. 2006...

Ans. :

1) In general, the memory management modules performs following functions :

a) To keep track of all memory locations free or allocated and if allocated, to which proc... and how much.

b) To decide memory allocation policy i.e. which process should get how much memo... when and where.

c) To use various techniques and algorithms to allocate or deallocate memory locatio... Normally, this is achieved with the help of some special hardware.

2) The following are the memory management systems :

A) Contiguous, Real Memory Management System :

(a) Single contiguous (b) Fixed partitioned

(c) Variable partitioned

B) Non - contiguous, Real Memory Management System :

(a) Paging (b) Segmentation

(c) Combined

C) Non - contiguous, Virtual Memory Management System :

(a) Virtual memory

Q. 43 Explain memory map of single user operating system.

(March 2004, 2007, 2011, 2020)

Ans. :

Free memory
Process
Command interpreter
Kernel

Memory map of single user OS

(1) The operating system like MS-DOS is single user O.S.

(2) The memory map of such operating systems consists of program to be executed i... process, free memory available, command interpreter and kernel.

(3) The command interpreter of the single user operating system is invoked when t... computer is started.

(4) This O.S. loads program to be executed in main memory and assigns as much mem... as possible to it. It then sets instruction pointer and executes the program.

(5) If the program is terminated, then it is removed from memory.

(6) The kernel of such O.S. provides basic operating systems services, while the comm... interpreter interpretes the commands.

Q. 44 What is partitioning ? Explain fixed and variable partitioning.

(Mar. 2004, 07, 08, 09, 12; Oct. 2005, 14, July 2017, 18)

Ans. : Certain operating systems use partitioned memory management to allow multiprogramming. Partitioning means dividing main memory into various sections. These sections are called partitions.

There are two types of partitions :

I) Fixed partitions II) Variable partitions

I) Fixed partitions (Static Partition) :

(1) In fixed partitioning, partitions could be of different sizes. But once decided, their size can not be changed.

(2) In this method partitions are fixed at the time of system generation. At this time, system manager has to declare the partition time.

(3) Fixed partitions are also called as static partitions. On declaring fixed partitions, the operating system creates Partition Description Table (PDT).

II) Variable partitioning :

(1) In variable partitioning number of partitions and their sizes are variable.

(2) They are not defined at the time of system generation.

(3) These partitions are created by the operating system at run time they differ in size.

(4) The procedure to be followed for memory allocation is nearly same as that in case of fixed partitions.

(5) At any time, any partition may be free or allocated to some process. Also, in variable partitioning, starting address of partition is not fixed.

Q. 45 Explain difference between fixed partition and variable partition.

Ans.:

Sr. No.	**Fixed Partition**	**Variable Partition**
1.	Partition created could not be changed.	Partition created can be changed.
2.	Partition can be defined at the time of generation.	Partition cannot be defined at the time of generation.
3.	It suffers from problem of internal fragmentation.	Problem of internal fragmentation is solved.
4.	No. of partition and their sizes are fixed.	No. of partition and their sizes are variable.

Q. 46 Give the disadvantages of fixed partition.

Ans. :

(i) Fixed partitioning suffers from internal fragmentation i.e. wastage of memory space within the partition e.g. suppose the partition size is 200k and a job is of 100k, then 100k of memory will be wasted.

(ii) Fixed partitioning reduces degree of multiprogramming.

(iii) It also restricts C.P.U. utilisation.

Variable partitioning overcomes these problems and hence it is widely accepted.

Q. 47 State the various steps involved in the allocation of a partition in case of fixed partition memory management. **(March 2002, 2005, 2006, 2007, 2017/Oct. 2003,10,12)**

Ans. : When a process is to be allocated a partition, following take place :

i) The long term process scheduler of the PM decides which process to be brought in to the memory.

ii) It then finds out the size of the program to be loaded by consulting the IM portion of the O.S. The compiler keeps the size of the program in the header of the executable file.

iii) Then makes a request to the partition allocation routine of the MM to allocate a free partition, with the appropriate size.

The partition description table (PDT) is useful for this procedure.

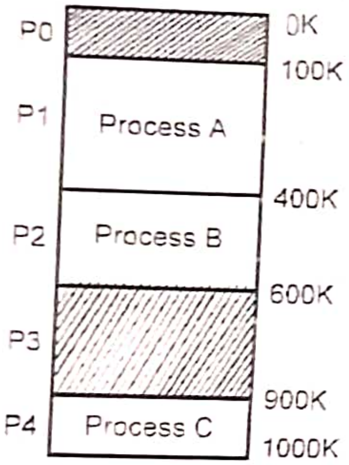
iv) With the help of the IM module it now loads the binary program in the allocated partition.

v) It then makes an entry of the partition ID in the PCB (process control block) before the PCB is linked to the chain of ready processes by using the PM module of the operating system.

vi) The routine in the MM now redefines partition description table and marks the status o that partition as allocated. (ALLC)

vii) The PM eventually schedules this process.

A partition description table (PDT) is shown in Fig. below :



Fixed partition

Partition ID	Partion		
	Starting Address	Size	Status
0	0	100K	FREE
1	100K	300K	ALLC
2	400K	200K	ALLC
3	600K	300K	FREE
4	900K	100K	ALLC

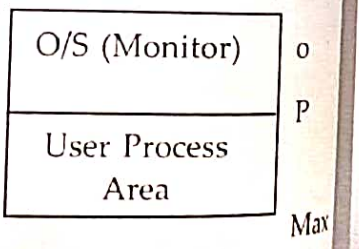
PDT

Q. 48 Explain the single contiguous memory management systems with a suitable memory mapping diagram. **(March 2005, Oct.2010)**

Ans. : Single contiguous memory management system :

(1) In this memory management system, the physical memory is divided into two contiguous areas. One is permanently allocated to the resident portion of the operating system and the remaining used for user process.

(2) As shown in figure operating system may be loaded at lower addresses i.e. 0 to P. At any time, only one user process is in the memory. This process is run to completion and then the next process is brought in the memory.



Q. 49 What is paging ? Explain in detail. OR **(March. 11, 19, Oct. 2004,15, July 2016)**
Explain page memory management system with a suitable page map Table (PMT) **(Oct. 2008)**

Ans. :

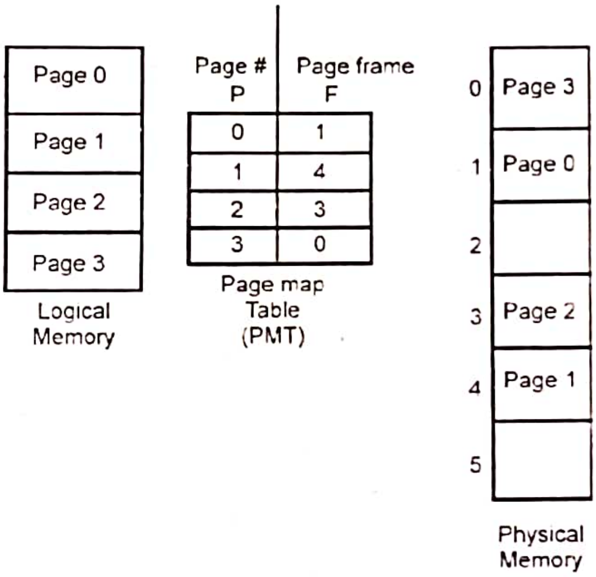
(a) Partitions suffer from external fragmentation because of available memory is not contiguous.

(b) Paging permits a program's memory to be non-contiguous allowing a program to be allocated physical memory wherever it is available.

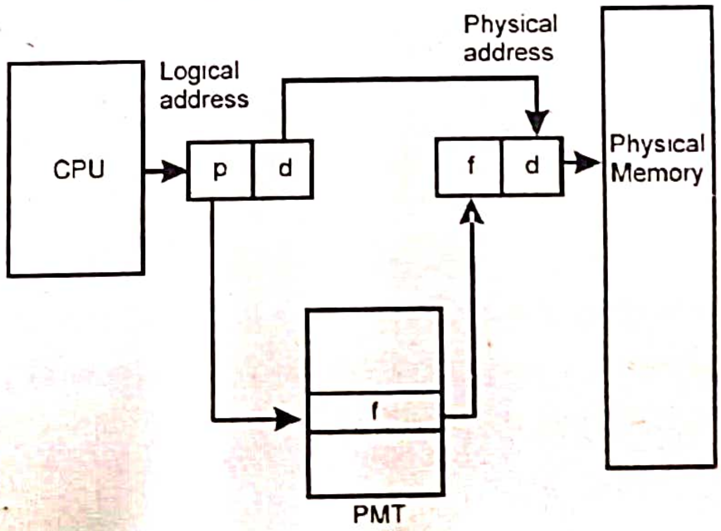
(c) Physical memory is broken into fixed-size blocks called Page Frames. Logical memory is also broken into blocks of the same size called Pages.

(d) When a program is to be executed its pages are loaded into any available frames and the page map tables defined to translate from user pages to memory frames.

(e) The page size is defined by hardware. It is typically power of 2.
The paging model of memory is shown as follows :



(f) Every address generated by CPU is divided in two parts :- a page number (p) and a page offset / displacement (d). The page number is used as an index into a PMT.

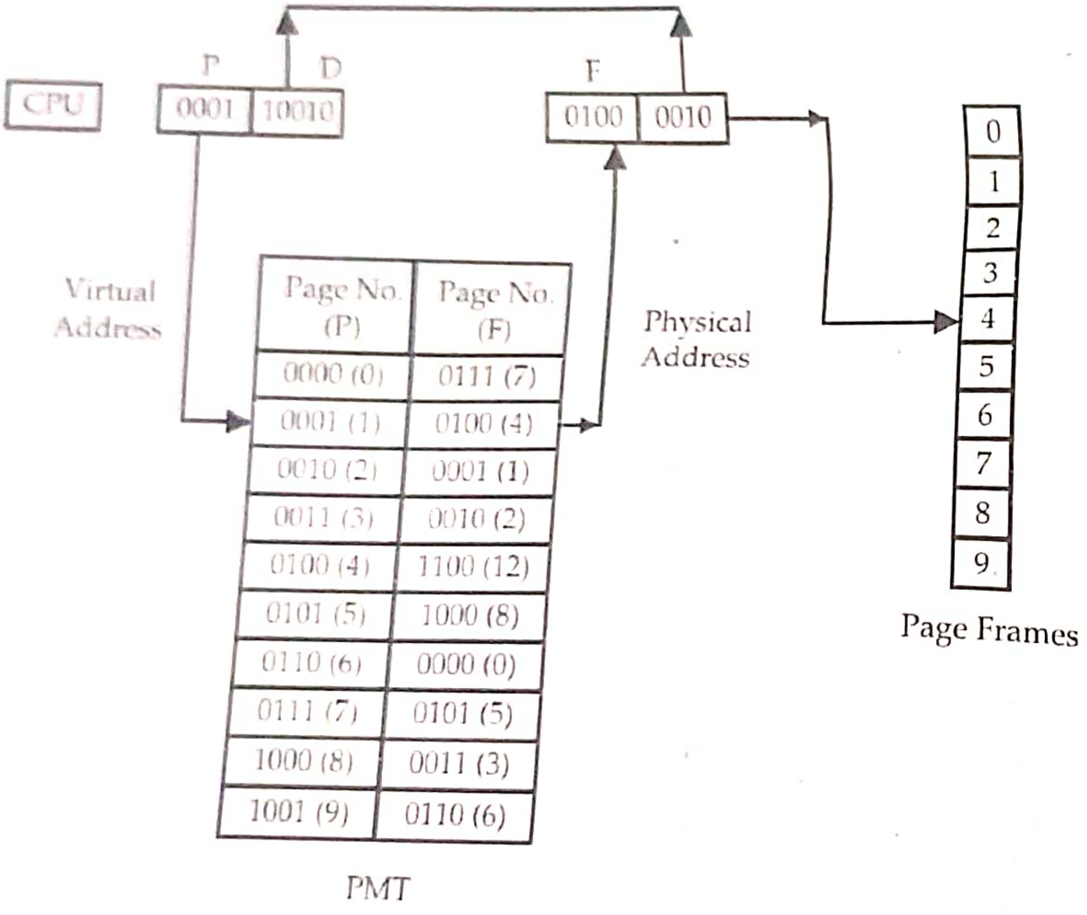


Q. 50 Explain address translation mechanism in paging?' Why is the page size normally some power of two?

Ans. :

Let us assume that we have' a machine where a word length is 8 bits. Let us also assume that our machine has main memory with capacity of 512 bytes or words. This memory is divided into 16 pages of 32 words each (16 x 32 = 512). Hence we require 4 bits to represent page number P (0 to 15) and 5. bits to represent displacement D (0 to 31). Therefore total number of bits in address will be 9 (4 + 5).

The address generated by compiler is divided into page number [P] and displacement I. If any instruction at virtual address 50 is be executed then compiler generate P = 1 and D = and in binary P = 0001 and D = 10010. But figure shows that page is mapped onto page time 4 and thus Physical address of instruction is in page 4 at displacement 18. Page form 4. It will contain physical address 128 + 18 = 146. Hence we need to fetch instruction not at location but at location 146.



Page No. (P)	Page No. (F)
0000 (0)	0111 (7)
0001 (1)	0100 (4)
0010 (2)	0001 (1)
0011 (3)	0010 (2)
0100 (4)	1100 (12)
0101 (5)	1000 (8)
0110 (6)	0000 (0)
0111 (7)	0101 (5)
1000 (8)	0011 (3)
1001 (9)	0110 (6)

Page size is normally some power of two because:

Complier generate only one dimensional single address in binary these address are divided into high order bit as P (Page number) and low order bit as D (Displacement in page) which is only happens if the page size is a power of 2. If the page size is not a power of 2, this automatic separation of P and D does not take place.

Q. 51 What is segmentation? **(March.2011, Oct. 2004)**

Ans. :

(1) Segments are logical divisions of programs and hence are normally of variable sizes.

(2) Segmentation is a memory management scheme which support user's view of memory.

(3) Each segment has number and length.

(4) Each program in executable form can be considered to be consisting of different segments such as code, data and stack. Each of these can be further divided into new segments.

(5) A program normally contains main program, some subprograms and few predefined and precompiled functions. Each of these belongs to different segments.

(6) An application programmer does not necessarily have to declare different segments in the program. If various segments in his programs does not define explicitly, then the compiler does it by its own. Following are the jobs of compiler :

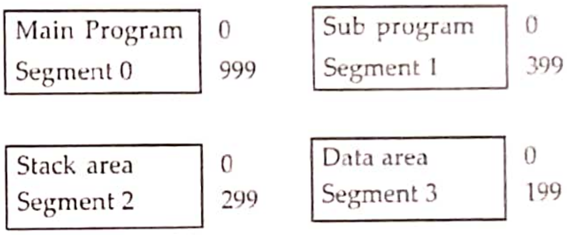
i) Recognize different segments in program.

ii) Number those segments.

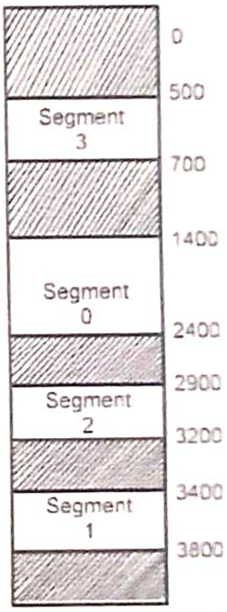
iii) Define segment table

iv) Produce an executable image by assigning two dimensional addresses.

(7) Consider the examples as shown in figure Four segments numbered 0 to 3. The SMT (Segment Map Table) has separate entry for each segment giving the size and base of segment.



Segments



Segment #	Limit	Base
0	1000	1400
1	400	3400
2	300	2900
3	200	500

Segment map table

Q. 52 Differentiate between Paging and Segmentation. **(Oct. 2011)**

Ans.:

Sr. No.	Paging	Segmentation
1.	Pages are physical in nature.	Segments are logical division of programs.
2.	Pages are of fixed. size. - .	Segments are of variable size.

Sr. No.	Paging	Segmentation
3.	Page table translate from user pages to memory frames.	The segment table has a separate entry for each segment in physical memory and length of that segments.
4.	In paging, logical memory is divided into physical memory.	In segmentation main program is divided into segments.

Q. 53 Explain the concept of virtual memory in brief. **(March 2004, 15, 16; July 2019)**

Ans. :

(1) Virtual memory is an attempt, which makes the execution of the processes possible which may completely not in the main memory. Some part of the process may be on disk.

(2) The MM techniques are simple to implement but the major drawback is that if the physical memory is limited then number of processes it can hold at any time i.e. degree of multiprogramming reduces. For this concept of virtual memory is introduced.

(3) The main advantage of virtual memory is user can execute programs, whose size may be greater than the physical memory.

(4) Virtual memory is difficult to implement. It can be implemented by paging segmentation or combined schemes. Mostly virtual memory systems are implemented by using paging.

(5) A program consists of number of logical or virtual pages. To start execution of program some of the pages are loaded into specific page frames. If a page is not loaded into memory and a location from that page is referenced, at that time a page fault arises. When a page fault arises the O.S. loads the referenced page in memory from disk.

(6) Generally virtual memory is related to following concepts :

(i) **Locality of reference :** In locality of reference it is forecast whether a page is likely to be referenced in near future or not depending on its behavior in past and hence may be removed from memory.

(ii) **Page fault :** When a page, which is not in main memory is referenced, then a page fault arises. At this time the O.S loads that page in main memory.

(iii) **Working set :** The set of pages in physical memory which are actively referred to any moment is called as working set. Working set helps to decide page replacement policy.

(iv) **Page replacement policy :** If there is no page frame in main memory to accommodate new page, the O.S. overwrites some existing pages. These pages are determined by page replacement policy.

(v) **Dirty page :** A page which is modified after it is loaded in main memory is called as dirty page.

(vi) **Demand paging :** In demand paging, a page is loaded in memory, only when it is demanded.

Q. 54 Explain the following terms. **(Oct. 2007, 14)**

Ans. :

(1) **Locality of reference :**

i) The basic principle behind virtual memory is called locality of reference.

ii) This gives some basis to forecast whether a page is likely to be referenced in the near future or not, depending on its behavior in past.

iii) Thus, it helps to decide whether the page should be thrown out from main memory to make space for new page or not.

iv) If this principle is not valid, user can not throw any pages out on the disk from memory without a possible severe degradation in the performance.

2) **Page fault :** **(March.2011, Oct. 2006)**

i) In many systems, when a process is executing with only a few pages in memory and when an instruction is encountered which refers to any instruction or data in some other page which is outside the main memory i.e. on the disk, la 'page fault' occurs.

ii) At this stage, the operating system must bring the required page into the memory before the execution of that instruction can restart.

3) **Working set :** **(March.2011, Oct. 2006)**

i) At any time, a process has a number of pages in the physical memory. Not all of these are actively referred.

ii) The set of pages in the physical memory actively referred to at any moment is called working set.

iii) This has a significant bearing on the policy of bringing in pages from the disk to the main memory, if the operating system follows the "Working set model."

4) **Page replacement policy :** **(March.2011, Oct.-2006)**

i) As the number of processes and the number of pages in the main memory increase all the page frames become occupied.

ii) At this time, if a new page is to be brought in, the operating system has to overwrite some existing page in the memory.

iii) The page to be overwritten is selected by page replacement policy.

iv) There are a number of ways in which the O.S. selects the page to be overwritten. The O.S. designer chooses one amongst many of such policies and writes corresponding algorithm for it.

5) **Dirty page / Dirty bit :** **(March.2011, Oct. 2006)**

i) The page which is modified after it is loaded in main memory from disk is called as dirty page.

ii) The operating system maintains one bit for each physical page frame to denote whether a page has become dirty or not. This bit is called dirty bit.

6) **Demand paging :**

i) In demand paging, a page is brought in only when demanded.

ii) Consider a process is created with no pages in main memory. When the process is dispatched initially, the program counter will have been loaded with the address of first instruction. This address obviously belongs to a page outside the main memory. So a page fault will occur and the O.S. will now bring that page in memory.

iii) In this way as page fault goes on occurring, the O.S. brings new referred pages. This is called demand paging.

iv) The drawback of demand paging is that a lot of pages which have been used in past, but which now are not required, remain in memory unnecessarily.

Q. 55 What is dirty page, dirty bit? How are they used?

Ans.:

If the page in memory has been modified after it was loaded from the disk is called page.

The operating system maintains one bit for each physical page frame to denote whether a page has became dirty or not. This bit is called 'dirty bit'.

The hardware is normally designed in such a fashion that if a page is modified, the dirty bit for that corresponding page frame is set to 1 automatically; otherwise, it is at 0.

If a page is dirty, it cannot be blindly overwritten, because otherwise the updates will be lost. Hence in such a case, this page has to be written back onto the disk at a proper place, so that when it is brought in next time, the correct and current copy is brought in. If the page is not modified i.e. clean, the operating system need not write it back. In this way, a lot of time is saved.

GUI

Q. 56 What is GUI ? State any four advantages of GUI. **(Oct. 2003, 2006, 2007, July 2017**

Ans. :

GUI : The interface which replaces cryptic commands by their graphical representatio are called **Graphical User Interface (GUI)**.

Windows operating system is GUI based operating system.

Advantages of GUI :

1) With GUI, commands are replaced by graphics. Hence it is not necessary to remembe the command and its meaning.
2) With GUI, user can run several programs simultaneously.
3) User can communicate and exchange data between programs without transferring or copying files.
4) Easy to use, consistent GUI for virtually all programs.

Q. 57 What is GUI ? Explain in brief any two features of GUI. **(March 2005, 2012 Oct. 2006)**

Ans. :

GUI : The interface which replace cryptic commands by their graphical representation are called Graphical User Interface (GUI).

Windows operating system is GUI based operating system.

Features of GUI :

1) **Replacement of command with icons :**
   - (a) Commands are grouped together in various levels of hierarchy and when the user selects a group, further commands in that group are displayed.
   - (b) This allows the user to select a command using a cursor and simply clicking on it.
   - (c) User can select the command and use the application without first having to know about the computer and its working.
   - (d) The display of these command sequences takes place graphically.

2) **Provide on-line HELP :**

(a) GUI – based applications provide "HELP" about various features of the application.

(b) HELP can assist the user in knowing everything about the application.

(c) In the windowing environment, if user gets confused at any point, a HELP is readily available. This makes GUI-based applications more popular and efficient.

Q. 58 What is meant by GUI ? What are the essential components of GUI ? Explain any three. **(March 2004, 18, Oct. 2004, 12, 15, July 2018, 19)**

Ans. : GUI : The interface which replace cryptic commands by their graphical representation are called Graphical User Interface (GUI).

Windows operating system is GUI based operating system.

Essential components of GUI are :

(1) Menu bar

(2) Scroll bar

(3) Controls push button, option button, radio button, check box, list box, Entry box, Combo box)

(4) Dialogue boxes

(5) Feedback pointer

1) **Menu bar :**

(i) Menu bar normally appears at the top of the window under the window title.

(ii) Some commonly used menu bar options are File, Edit, View, Help etc.

(iii) When one of these menu is selected, a **pulldown menu** appears on the screen.

2) **Scroll bar :**

(i) Scroll bars allow user to scroll window horizontally and vertically.

(ii) Scroll bars are generally used to look at information, which is not currently visible in screen, by scrolling window horizontally or vertically.

(iii) A scroll bar consists of a horizontal or vertical scroll area with a slider box and an arrow in a box at each end.

vi) Slider box gives a hint on size and position of the visible part of object.

3) **Dialogue boxes :**

(i) Dialogue box is a window, used by the application to interact with the user.

(ii) A dialogue box can also be used to display information or to get user input and also for a combination of these two functions.

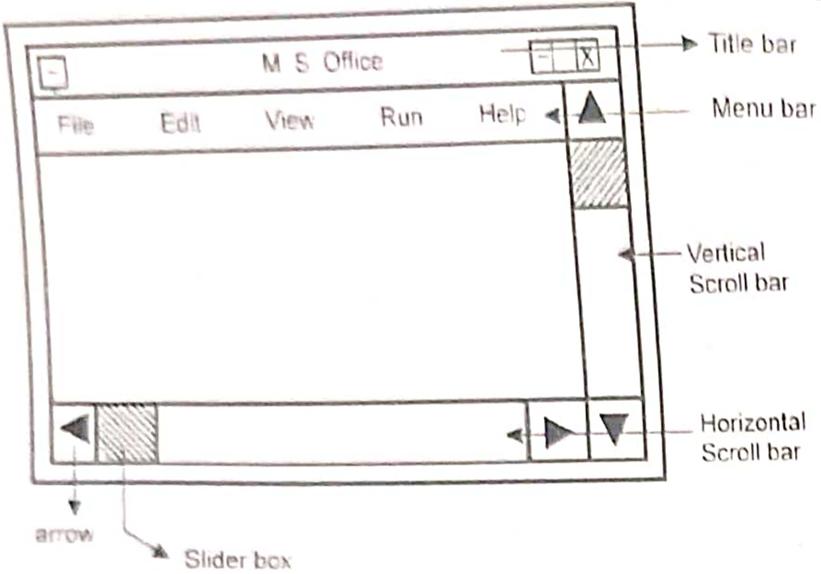
(iii) Dialogue boxes are of two types : (i) Modal dialogue box (ii) Modeless dialogue box.

(iv) A dialogue box where an application can continue only after the user has responded to the dialogue is called modal dialogue box.

(v) A dialogue box , which allows user to continue without responding to it is called modeless dialogue box.

Q. 59 Explain in short the function of Menubar and scroll bar components of G.U.I. **(March 2002, 2008, 2011)**

Ans. :



(i) **Menubar :**

i) A menubar consists of different main menus, which can be used in program.

ii) The main menus consists of different submenus. A menu can be selected by clicking it.

iii) A menubar is normally present at the top of the window under the window title. Some of the commonly used menu options are File, Edit, Help etc.

iv) When one of these menu is selected by clicking, a pull down menu list appears on the screen.

v) A pull down menu is a rectangular box, with more specific action listed in the box, out of which one can be selected by clicking a particular menu item.
e.g. In pull down menu of "file"
(i) Open : opens required document by giving browsing window.
(ii) Print and Save : prints and saves current document.

(ii) **Scroll bars :** **(March 2007)**

i) Scroll bars allow user to scroll window horizontally and vertically.

ii) Scroll bars are generally used to look at information, which is not currently visible in screen, by scrolling window horizontally or vertically.

iii) A scroll bar consists of a horizontal or vertical scroll area with a slider box and an arrow in a box at each end.

iv) Slider box gives a hint on size and position of the visible part of object.

Q. 60 Explain the following with respect to GUI : (ii) Title bar. **(March 07, March 18)**

Ans. :

1. *Title bar appears at the top of the window.*
2. *Title bar shows the title of each window.*
3. *The title helps to identify each window separately*

Q. 61 Explain various controls of G.U.I.

Ans. : A variety of controls are used in a G.U.I. to enable user to select type of information or to select specific operation to be carried out. That are either buttons or boxes. Some of the control buttons are :

1) **Push button :**

It is a rectangular button having a label, indicating action to be carried out. This button is used to select an action represented by button. This buttom hormalhy used when one action is to be selected at of many choices.

For e.g. SAVE

2) **Option button or radio buttons :**

It consists of a graphical image, which is used to select one object out of several possible objects. The currently selected can be distinguished from the others by highlighting on the graphic image :

e.g. ○ Workarea

○ Modeless

○ Application.

3) **Check buttons :**

A check button consists of a square box and an accompanying text. This is used for selecting one or more choices from a list of options.

e.g. ☑ Auto manager

☑ Default position

□ Align

4) **The Box controls are list box and entry box :**

(i) **List box :**

A list box is a rectangular box with scroll bars. This allows user to select one item from a scrollable list of choices.

(ii) **Entry box :**

Entry box is a rectangular box, which allows user to enter some text. An additional hint about the type of text to be entered is provided near the box.

Q. 62 What are dialogue boxes ?

Ans. :

i) Dialogue box is a window, used by the application to interact with the user.

ii) A dialogue box can also be used to display information or to get user input and also for a combination of these two functions.

iii) Dialogue boxes are of two types : (i) Model dialogue box (ii) Modeless dialogue box.

iv) A dialogue box where an application can continue only after the user has responded to the dialogue is called model dialogue box.

v) A dialogue box, which allows user to continue without responding to it is called modeless dialogue box.

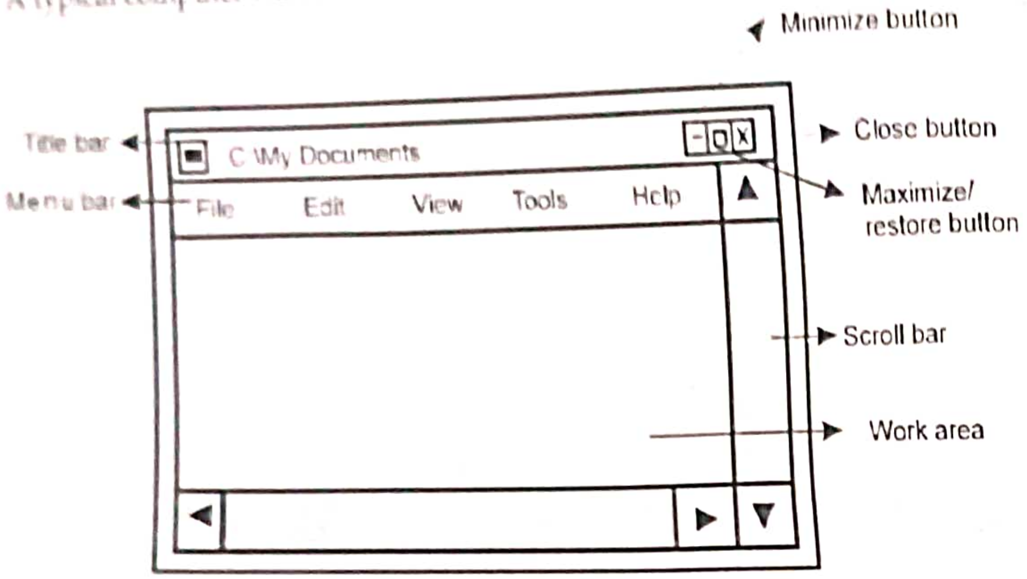
Q. 63 What is Windows ? What are the operations that can be performed on a window ? **(Oct. 2005)**

Ans. : Different applications are shown on computer screen by icons. User can open one or more applications at a time by clicking them.

The computer screen is divided into different partitions. Each partition can be of different size. User run different application in each partition of the screen. Each of these partition is called a window.

Windows are independent of each other.

A typical computer window is shown in figure below :



Following operations can be carried out on a window :

(i) **Dragging the window :** **(Oct. 2005)**

1) The position of the window on screen can be changed by dragging it.

2) To drag a window select a window by clicking mouse, keep the left button of the mouse pressed with mouse pointer on the tittle layer of windows.

e.g. on the layer in which c:\My documents is written in above example.

3) Move the mouse pointer to new position, it will find that the window is shifted to new position.

(ii) **Resizing window :** **(Oct. 2005)**

Making change in the size of window is called resizing window.

A window can be resized as follows :

1) Move the mouse pointer to right/left side border of windows.

2) Observe that the shape of mouse pointer gets changed to left/right arrow.

3) Now press the left button of mouse and move the mouse arrow to right/left side to a new position. Observe that right/left border of window is moved.

4) Similarly we can move top/bottom border of window.

Thus we can resize the window.

(iii) **Minimize/Maximizing Window :** **(March 2007, Oct. 2005, 2008, July 2016)**

We can minimize/maximize or close a window by pressing minimize/maximize or close the button respectively which are present at the top in the right corner of the window.

Q. 64 Explain in brief the following programs of MS-Windows :
(i) Program Manager (ii) File Manager
(iii) Control Panel **(March 2003, 2007, 2009, 2013)**

Ans. : MS-Windows environment provides following programs which play very important roles.

1) **Program Manager :**

(a) The Program Manager starts executing along with MS-Windows.

(b) This provides user interface to start and stop applications.

(c) It is used to organize various applications into different groups.

(d) It also indicates how each group contents are controlled and displayed on the screen.

(e) It is also used to end the MS-Windows session.

2) **File Manager :**

(a) This helps organize user files and directories.

(b) This is used to traverse through the file system and change drives, to search, copy, move, create or delete files and directories.

(c) Applications can be started directly from the File Manager.

3) **Control Panel :**

(a) It can be used to choose or change the color schemes in the applications, select and display the background of the screen, select border width and other border characteristics, cursor size and shape etc.

(b) Fonts also managed by controlpanel.

(c) It is also used to configure printers and other ports on the PC.

Access and Security Aspects of O.S.

Q. 65 Define "security" with respect to an operating system. Explain the different elements of security? **(Oct. 2002, 2004, 2010, 2012, March 2018)**

Ans. :

1) Security is concerned with the ability of the operating system to enforce control over the storage and transportation of data in and between the objects, that the operating system supports.

2) In multiuser operating systems, the concepts of security and protection are very important. User programs should not interfere with one another or with the operating system.

3) In general, Secure Systems are those, which control, through the use of specific security features, access to information that only properly authorized individuals or processes operating on their behalf will have access to read, write, create or delete.

4) There are three main elements of security viz. Confidentiality, integrity and availability.

(i) **Confidentiality :** **(March 2005; Oct. 2006, 2008; July 18)**

Confidentiality ensures that information is not accessed in an unauthorized manner. It is generally related to the Read operations.

(March 2005; Oct. 2006, 2008)

(ii) **Integrity :**

Integrity ensures that the information is not amended or deleted by an unathorised manner. It is generally related to Write operations.

(March 2005; Oct. 2006, 2008)

(iii) **Availability :**

It ensures that information is available to the authorised users at right time.

Q. 66 Discuss in brief threats to security in any computing environment.

Ans. :

1) Sharing and protection are requirements of modern computing environments. But these two are contradictory to each other as more sharing gives rise to possibility of more security threats.

2) The major threats to security in any computing environment can be categorized as follows :

   (i) **Tapping :** Unauthorised use of servicing.

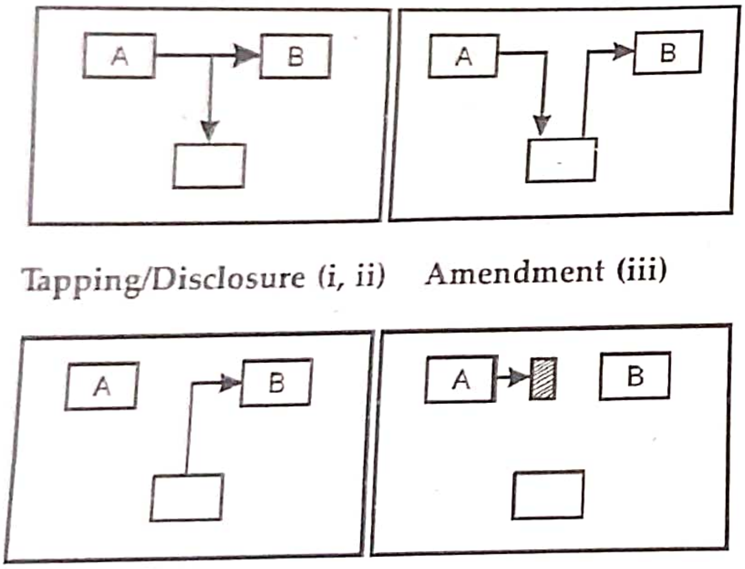
   (ii) **Disclosure :** Unauthorised disclosure to information.

   (iii) **Amendment :** Unauthorised alteration or deletion of information.

   (iv) **Fabrication :** Unauthorised fabrication of information.

   (v) **Denial :** Denial of service to the authorized users.

3) The security threats are shown in the following figure.



Tapping/Disclosure (i, ii) **Amendment (iii)**

Fabrication (iv) **Denial (v)**

4) Out of these five security threats, the first two, viz. tapping and disclosure, are categorized as passive threats and the other three as active threats.

5) It is clear that in both the cases (i) and (ii), information goes to the third party. But, the difference is that in tapping the third party accessed information without knowledge of other two parties. Whereas in disclosure the source party willingly discloses information to the third party.

Q. 67 What are attacks on security ? Explain in short the ways in which a system can be attacked. **(Oct. 2005, July 2016, 18)**

Ans. : The security system can be attacked and penetrated in a number of ways as follows :

1. **Authentication :**

   Authentication means verification of access to the system resources. Following are some of the ways in which authentication may take place :

   (i) By stealing and using somebody else's password and then use it.

   (ii) Use of vendor supplied password which can be used by only system administrator.

   (iii) Finding password by trial and error (i.e. guess) method.

   (iv) If a user logs on and then goes off, an intruder can use that terminal.

   (v) Writing dummy login programs to fool the user.

2. **Browsing :**

   (i) In some systems, there exist files with access controls, which are very permissive.

   (ii) One can browse through the system file to get this information, after which, unprotected files/databases could be easily accessed.

   (iii) Confidential information could be read or even modified.

3. **Trap doors :**

   (i) Sometimes, software engineers leave some secret entry point to modify their programs. These are called trap doors.

   (ii) They can be misused by others.

4. **Electric data capture :**

   Use of active or passive wire traps, or mechanism to pick up the screen radiation and to recognize what is displayed on screen is called electric data capture.

5. **Invalid parameters :**

   Passing invalid parameters may cause serious security violations.

6. **Line trapping :**

   A special terminal is used to tap into a communication line. It causes access to confidential data.

7. **Waste recovery :**

   By using some technique, deleted files can be recovered, password may be recollected.

8. **Rouge software :**

   Certain programs like worms, viruses attack on system.

Q. 68 What are computer worms ? Explain its mode of operation. **(Mar. 2003,13 Oct. 2010, 13)**

Ans. :

1. A computer worm is a complete program by itself. It is written in such a way that it spreads to other computers over a network.
2. But, while doing this, it consumes the network resources to a very large extent.
3. A computer worm can potentially bring the entire network to a grinding halt.

Mode of Operation :

4. Usually a computer worm does not harm other programs or data.
5. It just spreads, thereby consuming large resources such as transmission capacity or disk storage. It denies services to legitimate users.
6. A computer worm usually operates on a network. Each node on network maintains a "mailing list", which contains the names and addresses of the reachable machines on the network. The worm gets access to this list and using this, sends a copy of itself to all those addresses.
7. If the worm is intelligent, after reaching a node it checks whether a copy of itself already exist there or not. If exists, it does not create one more copy.
8. If the worm in dumb, it just copies itself to all nodes. So, if one node's address is at several places in network, then it would have several copies of the worm.
9. **Safe guards against worms :**

   A worm can be prevented by strong security and various check points on the communication system.

Q. 69 What is a computer virus ? State various types of viruses and the basis on which they are classified. **(March 2002, 06, 07, 11, 17, 20, Oct. 2007,15, July 17, 19**

Ans. :

1. A computer virus is a part of program, which is written with clear intention of infecting other programs.
2. A computer virus is not a complete program by itself. It can not act independantly.
3. A computer virus causes direct harm to the system. It can corrupt code as well as data.
4. The classification of virus is based on what do it affects or where the virus resides.
5. There are five types of viruses given below :

   (i) Boot sector virus   (ii) Memory resident virus

   (iii) File specific virus   (iv) Command processor virus

   (v) General purpose virus.

Q. 70 What are the different methods by which virus can infect other programs ? **(March 2005, 2011, 2017, Oct. 2007,15, July 2017, March 2020)**

Ans. : There are five well known methods by which a virus can infect other programs :

(i) **Append :**

In this method the viral code appends itself to the unaffected programs.

(ii) **Replace :**

In this case, the viral code replaces the original executable program completely or partially to carry out some funny actions.

(iii) **Insert :**

In this case, the viral code is inserted in the body of an executable code to carry out some funny or undesirable actions.

(iv) **Delete :**

In this case, the viral code deletes some code from the executable program.

(v) **Redirect :**

This is an advanced approach employed by the authors of sophisticated viruses. The normal control flow of a program is changed to execute some other code, which could exist as an appended portion of normal program.

Q. 71 How generally a virus operates ?

Ans. :

1. A virus works in a number of ways. Normally, the developer of a virus has to be a very bright person who knows the operating system very well in order to break it.
2. This person produces interesting or useful program such as a good game or utility. However, this program has some viral code embendded in it.
3. Typically, it is developed under MS-DOS, as viruses are very popular on the PCs.
4. This program is then published on the public bulletin board system or it is distributed to people free of charge.
5. Tempted by its contents and the price, the user acquires it and then starts using it after copying it onto the machine.
6. At this stage, the virus can be said to be in a nascent state. After executing the game or the utility i.e. the host program, the virus also executes, which allows it to spread to other programs on the machine and infect them.

Q. 72 Discuss virus detection, removal and prevention philosophies.

(Mar. 2008,11,16, 17,19; Oct. 2003,05,13,July 2016)

Ans. :

(i) Virus detection :

1. Normally, a virus detection program checks the integrity of the binary files.
2. The program maintains a check sum on each file. A mismatch in it indicates virus.
3. Some programs reside in the memory and continuously monitor certain memory and I/O operations for guarding against any suspicious behavior.

(ii) Virus removal :

1. A generalized virus removal program is very difficult to imagine due to the multiplicity of the viruses and the creativity with which they are constructed.
2. However, for some viruses, bit pattern in code can be predicted.
3. In this case virus removal program scans the disk for the patterns of known viruses. On detection, it removes them. But, if the virus has already damaged data, then recovery of data is almost impossible.

(iii) Virus prevention :

1. User cannot cure the data (recover) after viral affection. Hence the best way is to prevent viruses.
2. For this user must buy official, legal copies of software from reliable stores or sources.
3. One should be extremely careful about picking up free, unreliable or illegal software.
4. Frequent back-ups and running of monitoring programs also help in detection, and thus subsequent prevention of different viruses.

Q. 73 Differentiate between computer worms and computer viruses.

(March 2004,09,12,14,15,19; July 18

Ans. : Computer worms :

i) A computer worm is a complete program.

ii) A computer worm can act independently.

iii) Generally it do not cause direct harm to the computer system.

iv) It just goes on spreading on to network and consumes network resources to a large extent.

Computer viruses :

i) A computer virus is not a complete program, but a part of program.

ii) A computer virus can not act independently.

iii) It causes direct harm to the computer system. It has been written with clear intention of infecting others.

iv) A computer virus corrupts code and data.

Q. 74 What is the difference between a Worm and a Virus ? Explain how these can be prevented.

(March 200

Ans. : Difference between a Worm and a Virus : Refer Q. No. 73.

1) **Worm Prevention :** A worm can be prevented by strong and various check points of communication system.

2) **Virus Prevention :**

(a) Users cannot cure the data after viral affection. Hence the best way is to prevent viruses.

(b) For this user must buy official, legal copies of software from reliable stores or sources.

(c) One should be extremely careful about picking up free unreliable or illegal software.

(d) Frequent back-ups and running of monitoring programs also help in detection, and thus subsequent prevention of different viruses.

Q. 75 Select the correct alternative and rewrite the following.

(March 2003, Oct. 2002

1. Operating system is ———

   1) hardware 2) software

   3) input device 4) output device

Ans. : (2) Software

2. ——— is service in operating system.

   1) Information management 2) Process

   3) G.U.I. 4) None of these

Ans. : (1) Information management

3. Windows NT is ——— operating system.

   1) Single user multitasking 2) Multiuser multitasking

   3) Time sharing 4) None of these

Ans. : (2) Multiuser multitasking

4. Linux is a —— software. **(March 2012, March 2020)**

1) public domain 2) free
3) paid 4) private

Ans. : (2) free

5. Windows 98 is ——— operating system.

1) single user multitasking 2) Multiuser
3) Time sharing 4) Multithreading

Ans. : (1) single user multitasking

6. The time required for read-write head to move to the correct track is ——— **(March 2009)**

1) Seek time 2) Rotational delay
3) Latency time 4) None of these

Ans. : (1) Seek time

7. Termination of a process is done by ——

1) Memory management 2) Process management
3) Device driver 4) Information management

Ans. : (2) Process management

8. The time lost in turning the attention of processor from one process to other is called as ____________. **(March 2017)**

1) Circuit switching 2) Band width
3) Context switching 4) None of these

Ans. : (3) Context switching

9. —— is a function of memory management.

1) Creation of file 2) Halting process
3) Paging 4) None of these

Ans. : (3) Paging

10. If the page size for 2 MB memory is 2 kB, then the number of higher order bits on address bus, used to denote page number is —— **(March 2002, March 2018)**

1) 11 2) 10
3) 9 4) 8

Ans. : (2) 10

11. Wastage of memory space within the partition is called as ——

1) Internal fragmentation
2) External fragmentation
3) Compaction
4) None of these

Ans. : (1) Internal fragmentation

12. If a page is modified after it is loaded in main memory, then it is called as ——

1) Page fault 2) Dirty page
3) Paging 4) Locality of reference

Ans. : (2) Dirty page

13. Pages are physical in nature, while segments are ——
1) logical 2) virtual
3) either physical or logical 4) either virtual or physical

(March 2006)

Ans. : (1) logical

14. Following is not a process state —
1) ready 2) blocked
3) resumed 4) running

(Oct. 2003)

Ans. : (3) resumed

15. Windows NT is ——— operating system.
1) Multiuser 2) Multitasking
3) Multithreading 4) All of the above

(March 2004, 14)

Ans. : (4) All of the above

16. ——— is not an operating system.
1) UNIX 2) LINUX
3) MS-DOS 4) C++

Ans. : (4) C++

(Oct. 2004)

17. ——— spread more rapidly but causes less damage to computer networks.
1) Virus 2) Worms
3) Bombs 4) None of these

(March 2005, July 17)

Ans. : (2) Worms

18. ——— are the operating system programs.
1) Application program 2) User program
3) Process management program 4) Antivirus program

Ans. : (3) Process management program

19. ——— is given to each process so that a process does not use the CUP indefinitely.

(Oct. 2005)

1) Context Switching 2) Time Slice
3) Token Time 4) Purchased Priority

Ans. : (2) Time Slice

20. In Information Management _____ service is provided by operating system.

(Oct. 2006)

1) Change the priority of process 2) To allocate a chunk of memory to process
3) Open a file (for read, write or both) 4) Wait for a child process to terminate

Ans. : (3) Open a file (for read, write or both)

(March 2007)

21. _________ is an operating system.
(i) C++ (ii) C (iii) VB (iv) LINUX

Ans. : (iv) LINUX

(Oct. 2007)

22 Scheduling Policy is the term related to _____ Management of O.S.
i) Information ii) Process iii) Device iv) Memory

Ans. : (ii) Process

23. Terminate a Process is the system call available in _____ management. **(March 2008)**

i) Process ii) Memory iii) Information iv) File

Ans. : (i) Process

24. _______ program is a part of operating system. **(Oct. 2008)**

(i) Application (ii) Antivirus (iii) Process Management (iv) User's

Ans. : (iii) Process Management

25. Data is instantly updated, in case of ______ Operating System. **(March 2010)**

(i) Batch Processing (ii) Multi Programming

(iii) Multi User (iv) Real Time.

Ans. : (iv) Real Time

26. _______ are not operating' system programs. **(Oct. 2010)**

i) Process Management Programs ii) Information Management Programs

iii) Memory Management Programs iv) Anti -Virus Programs

Ans. : (iv) Anti -Virus Programs

27. A program under execution, which competes for the CPU time and other resources is Called ________. **(March 2011)**

i) Process ii) Multi-programming iii) Context Switching iv) None of these

Ans. : (i) Process

28. It the page size for 1 MB is 2 KB then the number of higher order bit of the address bus used to denote the page number is ________ **(Oct. 2011)**

(i) 11 (ii) 10 (iii) 9 (iv) 8

Ans. : (iii) 9

29. 'Terminate a Process' is System call in ________ **(Oct. 2012)**

(i) Information Management (ii) Process Management

(iii) Memory Management (iv) None of these

Ans. : (ii) Process Management

30. If the page size of 1 MB Memory is 1 KB,, then the number of higher order bits of the address bus used to denote page number is ______ **(March 2013)**

(i) 10 (ii) 11 (iii) 12 (iv) 9

Ans. : (i) 10

31. If the page of 4MB Memory is 1KB, then the number of higher order bits of the address bus used for page number is ________. **(Oct. 2013)**

(i) 11 (ii) 21 (iii) 12 (iv) 22

Ans. : (iii) 12

32. ____________ is a free software. **(Oct. 2014, March 2019)**

(i) UNIX (ii) DOS (iii) LINUX (iv) WINDOWS

Ans. : (iii) LINUX

33. The time required to move R/W head to the particular track is called ________. **(March 2015)**

(i) Latency time (ii) Seek time
(iii) Waiting time (iv) Response time

Ans. : (ii) Seek time

34. ________ is not a process state in OS. **(Oct. 2015)**

(i) Ready (ii) Running (iii) Exited (iv) Blocked

Ans. : (iii) Exited

35. ________ is an Operating System. **(March 2016)**

(i) VBSCRIPT (ii) UNIX
(iii) C (iv) BASIC

Ans. : (ii) UNIX

36. Context Switching is a term related to ________ Management. **(March 2016)**

(i) Process (ii) Memory (iii) Information (iv) Device

Ans. : (i) Process

37. In memory management system, where the part of the process image is in the main memory and the other part is on the disk, is known as ________. **(July 2016)**

(i) Segmented Memory Management System
(ii) Single Contiguous Memory Management System
(iii) Virtual Memory Management System
(iv) Combined Memory Management System

Ans. : (iii) Virtual Memory Management System

38. Turning attention of CPU from one process to another is called ________. **(July 2018)**

(i) reference (ii) Context switch
(iii) Process scheduling (iv) Multiprogramming

Ans. : (ii) Context switch

Q. 39 Terminate a process is the system call available in ______ management. **(July 2019)**

(i) Process (ii) Memory (iii) Information (iv) File

Ans. : (i) Process

❑❑❑

Chapter 2

DATA STRUCTURES

Scope of the Syllabus

Probable marks : 17

- Introduction to data structures.
- Data structure operations.
- Algorithmic notations.
- Control structures.
- Arrays-Representation in memory, Traversing, Deleting, Sorting, Binary search in an array, Pointer arrays, Records in memory using array.
- Linked list - Representation in memory
- Trees, Binary trees - Representing binary tree in memory.

INTRODUCTION TO DATA STRUCTURES

Q. 1 Define the following terms :

(i) Data :

Ans. Data are simply values or set of values.

(ii) Group items : **(March 2018)**

Ans. Data items which are divided into subitems are called as group items. e.g. Date may be divided into three subitems - day, month and year. So Date becomes group item.

(iii) Elementary items : **(March 2018)**

Ans. Data items which are not divided into subitems are called as elementary items. e.g. pincode number cannot be divided into subitems. So it is elementary item.

(iv) Entity : **(March 2018)**

Ans. An entity is something that has certain attributes or properties which may be assigned values.

The values themselves may be numeric or nonnumeric.

e.g. A Bio-data sheet mainly contains :

Attributes ⟶	Name	Age	Sex	Education
Values ⟶	Atul	22	M	B.E. (Computer)

(v) Field : **(March 2005, 2009,2016; Oct. 2007,2014; July 2019)**

Field is a single elementary unit of information representing an attribute of an entity.

(March 2005, 2009,2016; Oct. 2007,2014; July 2019)

(vi) Record :

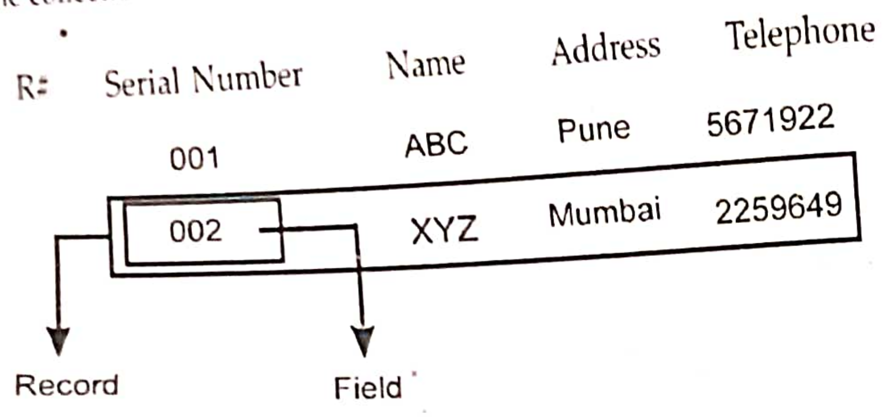
Record is a collection of field values of a given entity.

(March 2005, 2009,2016; Oct. 2007,2014 ; July 2019)

(vii) File :

File is the collection of records of the entities in a given entity set.

e.g.



R:	Serial Number	Name	Address	Telephone
	001	ABC	Pune	5671922
	002	XYZ	Mumbai	2259649

Q. 2 What is a data structure ? (Mar. 2006,2015; Oct. 2002, 2004, March 201

Ans. :

i) Data may be organized in many different ways. Data structure is the way in whi different data elements are logically related.

ii) Collection of data elements forming an organisation characterized by the accessi functions is called data structure.

iii) The data structure should be simple and it should show the relationship between da elements.

iv) **Types :**

(i) Linear data structure (ii) Non-linear data structure.

In **linear data structure,** data elements are stored in consecutive memory locations or b using linked representation. e.g. arrays, linked list.

In **non-linear data structures,** the linear order cannot be maintained between dat elements. Generally data elements have hierarchical relationship between them e.g. trees

v) Computer language provides different data structures like arrays, stack, queue, tree etc.

Data Structure Operations

Q. 3 Explain in brief any six data structure operations.

(Oct. 2002,04,06,12,15; Mar. 2002,06,12, July 2017, 19, March 2020)

Ans. : The data appearing in data structures are processed by means of certain operations like;

(i) Traversing :

Accessing each record or element exactly once, so that it can be processed is called as traversing.

For e.g. multiplying each element of an array by 6.

(ii) Inserting :

Adding a new record to the existing structure is called as inserting.

(iii) Deleting :

Removing a record from the existing structure is called as deleting.

(iv) Searching :

Finding the location of a record with given key values or finding the locations of all records which satisfy one or more conditions is called as searching.

(v) Sorting :

Arranging records in some logical order is called as sorting.

(vi) Merging :

Merging means combining the records in two different sorted files into a single sorted file.

Algorithmic Notation

(July 2017)

Q. 4 What is an algorithm?

Ans. :

i) An algorithm is a finite step by step list of well-defined instructions for solving a particular problem.

ii) An algorithm consists of two parts :

(a) First part is a **paragraph** which tells the purpose of algorithm. In this part, we define variables in algorithm and lists the input data.

(b) The second part of **algorithm** consists of steps in algorithm that are executed one after the another, generally beginning with step 1, unless stated otherwise. The control can be transferred to step n, by the statement "go to step n".

The algorithm is completed, when the statement 'Exit' or 'Stop' is encountered.

iii) e.g. Algorithm to find largest element in array.

Largest [DATA, N, MAX]

Here, DATA is a linear array with N elements. This algorithm finds the largest element MAX of DATA.

```
Step 1 : [Initialize counter]
         set k: = 1 and Max: = DATA [1]
Step 2 : [Compare and Update]
         If MAX < DATA [k + 1],
         then : MAX: = DATA [k + 1]
         [End of If structure]
Step 3 : [Increment counter] set k: = k + 1
Step 4 : [Test counter] If k < N, then : go to step 2
         [End of If structure]
Step 5 : Write : MAX
Step 6 : Exit
```

Q. 5 Write an algorithm to find smallest element in an array. **(March 2015, 3)**

Ans. :

Algorithm to find Smallest element in an array

Smallest[DATA , N, MIN]

Here DATA is a linear array with N elements. This algorithm finds smallest element NIN of DATA

```
Step 1 : [ Initialize counter ]
         set k = 1 and MIN = DATA[1]
Step 2 : [ Compare and.Update ]
         If MIN > DATA[ k +1]
         then MIN = DATA[ k +1]
         [ End of If Statement ]
Step 3 : Set k = k+1
Step 4 : If k < N then go to step 2
         [ End of If Statement]
Step 5 : Write MIN
Step 6 : Exit
```

Q. 6 Describe sequence logic or sequential flow.

Ans. :

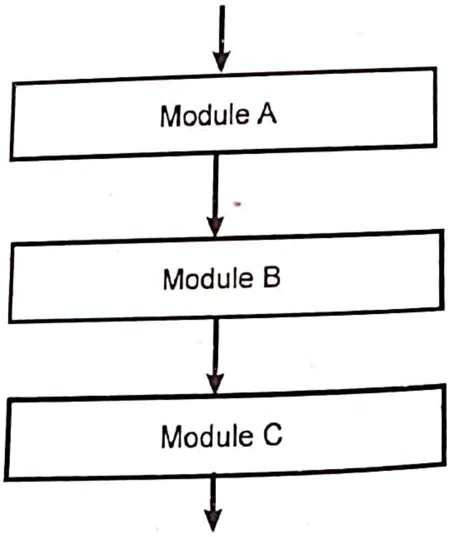
i) In the sequence logic, modules are executed sequentially, one after the another.

ii) The sequence may be present explicitly by means of numbered step or by the order in which modules are written.

iii) In short in sequential logic or sequential flow, modules of an algorithm are executed one after the another.

Algorithm
↓
Module A
:
:
Module B
:
:
Module C
:
:

Flowchart equivalent



Q. 7 Describe conditional flow or selection logic. **(July 2019)**

Ans. : Selection logic uses number of conditions, which cause selection of one out of several alternative modules. The structure which implement this type of logic is known as selection logic, or conditional structure.

There are three types of conditional structures :

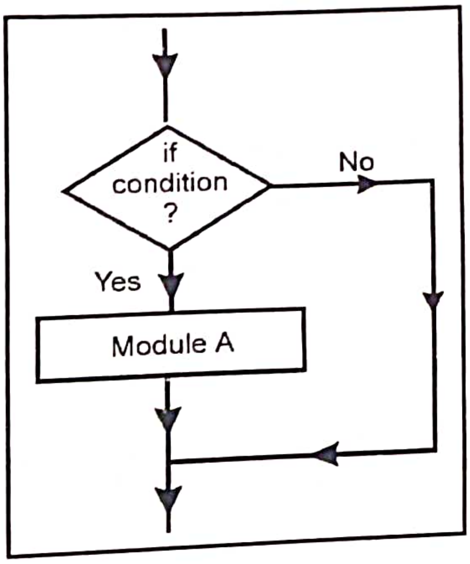
(i) Single alternative :

This has the form :

```
If condition, then :
        [module A]
[End of If structure]
```

The logic of this structure is as follows :

If condition is satisfied (true) then module A, which consists of number of statements, is executed. Otherwise, module A is skipped and next module of algorithm is executed.

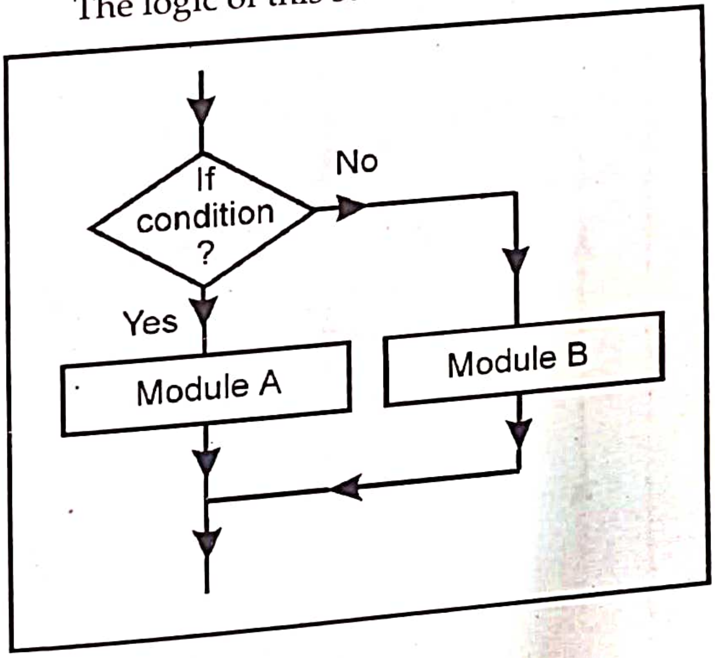


(ii) Double alternative :

This structure has the form :

```
If condition, then :
        [module A]
Else :
        [module B]
[End of If structure]
```

The logic of this structure is as follows



If the condition is satisfied, then module A will be executed other wise module B will be executed.

(iii) **Multiple alternative :**

This structure has the form :

```
If condition (1), then :
        [module A1]
else if condition (2), then :
        [module A2]
            :
            :
else if condition (n), then :
        [module An]
else :
        [module B]
[End of If structure]
```

The logic of this structure allows only one module to be executed. The module followi the condition, which is satisfied the condition will be executed. If no condition satisfied, then the module, which follows last Else statement will be executed.

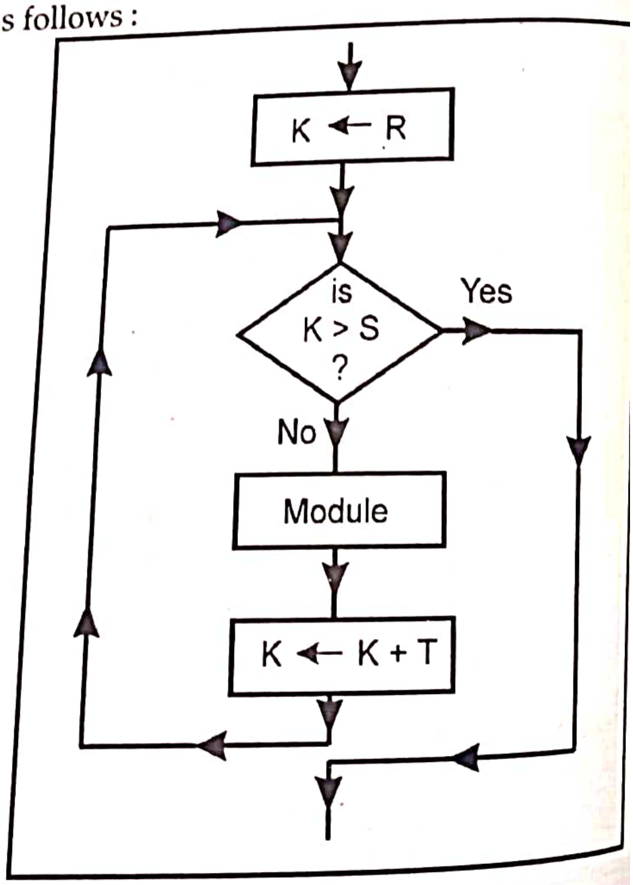
Q. 8 Describe logic of Repeat-For loop. (July 20

Ans. : The repeat-for loop has the form :

```
Repeat for K = R To S by T :
        [module]
[End of For loop]
```

Here, K is called index variable, R and S are initial and final values of K and T increment. The logic of this structure is as follows :

At first, the body of loop i.e. module will be executed with K = R and then with K = R + T, then with K = R + 2T and so on, until K < = S. The loop ends when K > S. If T is negative then K decreases in value and loop ends when K < S.

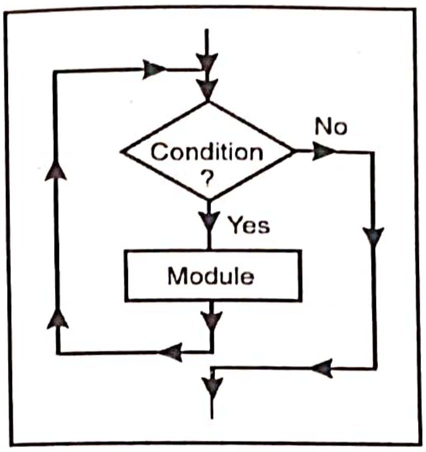


Q. 9 Explain Repeat-While structure.

Ans. : The repeat-while loop has the form :

```
Repeat While condition :
        [module]
[End of loop]
```

Here, body of loop i.e. module is executed repeatedly, unit the condition is satisfied.



There must be a statement before the structure that initializes the condition controlling the loop and there must be a statement in the body of the loop that changes the condition.

For e.g. Find largest element in array.

Given a nonempty array DATA with N numerical values. This algorithm finds the location LOC and the value MAX of the largest element of DATA.

```
1.    Set K := 1, LOC := 1, MAX : = DATA[1]
2.    Repeat step 3 and 4 while K < = N :
3.    If MAX < DATA [K], then :
          set LOC := K and
          set MAX := DATA[K]
      [End of If structure]
4.    Set K := K + 1
      [End of step 2 loop]
5.    Write : LOC, MAX
6.    Exit
```

Q. 10 Explain with flowcharts the following control structures : (i) Sequence logic, (ii) Selection logic, (iii)Iteration logic

(Mar. 09,12,17; Oct. 03, 04, 05,11,14; July 18, 19)

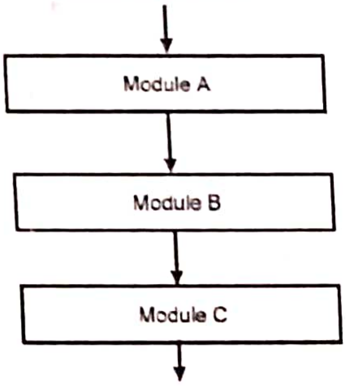
OR Explain 3 types of control structures used for flow of control.

Ans. :

(i) Sequence logic :

In the sequence logic modules are executed sequentially, one after the another. The sequence may be present explicitly by means of numbered step or by the order in which modules are written.

Flowchart equivalent

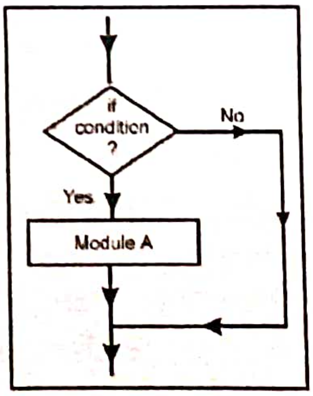


(ii) Selection logic :

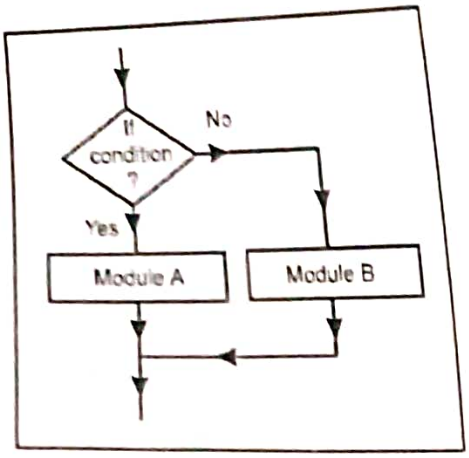
Selection logic uses number of conditions, which cause selection of one out of several alternative modules.

(a) Single alternative :

If condition is satisfied then module A, which consists of number of statements, is executed. Otherwise module A is skipped and next module is executed.



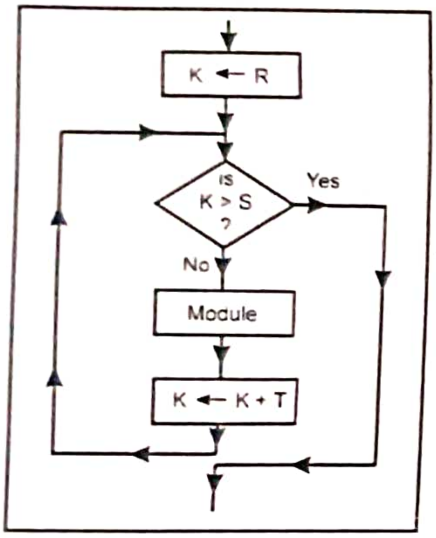
(b) Double alternative :



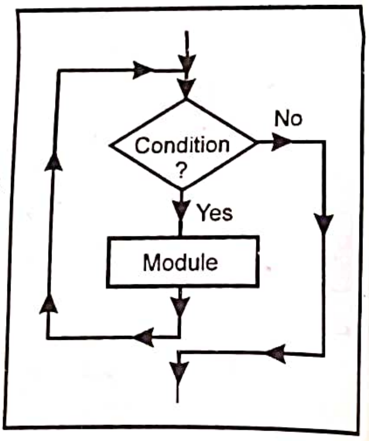
If the condition is satisfied, then module A will b executed otherwise module B will be executed.

(c) Iteration logic :

Here certain module is executed repeatedly until condition satisfies. At first, th body of loop i.e. module will be executed with K = R and then with K = R + T, then wit K = R + 2T and so on until K = S. The loop ends when K > S.



The repeat-while structure has the form :



Here module is executed until the condition is satisfied.

Q. 11 Write an algorithm to find solutions of quadratic equation $Ax^2 + Bx + C = 0$ wher $A \neq 0$ **(Oct. 2002, July 201**

Ans. : The algorithm inputs the coefficients A, B, C of a quadratic equation and outputs th real solution, if any.

```
1.  Read A, B, C
2.  Set D := $B^2 - 4 * A * C$
3.  If D > 0, then :
        a)  set X1 := $-B + \sqrt{D} / 2A$
            and X2 := $-B - \sqrt{D} / 2A$
        b)  Write : X1, X2
Else if D = 0, then :
        a)  set X := $-B/2A$
        b)  Write : 'UNIQUE SOLUTION' , X
```

```
Else
        Write : 'NO REAL SOLUTION'
[End of If structure]
4.      Exit
```

ARRAY

Q. 12 What are linear arrays ? **(Oct. 2006,13,14; Mar.2015, July 2016, 17, 19, March 18, 19)**

Ans. : A data structure is said to be linear if its elements form a sequence.

A **linear array** is the data structure which consists of finite, ordered set of homogeneous data elements such that :

1. The elements of **the array are referenced** respectively by an index set (subscript) consisting of 'n' consecutive numbers.
2. The elements of the array are stored respectively in successive memory locations.
3. The number 'n' of the elements is called length or size of array.

In general, the size or length of the array can be obtained from the index set by the formula :

$$\boxed{\text{Length} = \text{UB} - \text{LB} + 1}$$

where UB - the largest index called Upper Bound.

LB - the smallest index called Lower Bound.

e.g. Let DATA be 5 element linear array as follows :

	DATA
1	247
2	500
3	600
4	399
5	499

The element of an array may be denoted by the subscript notation such that :

DATA [3] = 600

In C++, array is declared as -

int data [100]; which specify an array data of 100 integers.

Q. 13 How arrays are represented in memory ? **(Oct. 2014)**

Ans. :

i) The elements of linear array are stored in consecutive memory locations.

ii) Computer does not need to keep track of the address of every element of array. It just requires the address of first element of array, LA, denoted by Base (LA) and called the base address of linear array LA.

iii) Using this base address, the computer calculates address of any element of array by using the formula.

$$\boxed{\text{LOC (LA[K])} = \text{Base (LA)} + \text{W (K} - \text{LB)}}$$

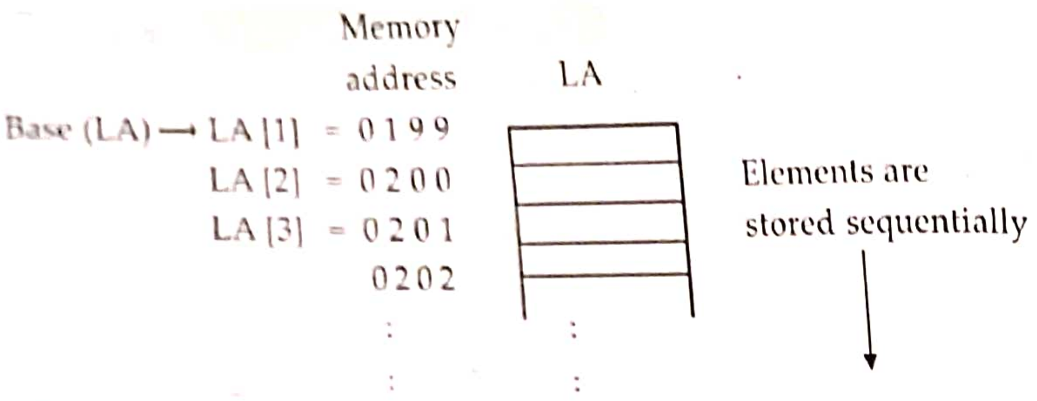
where,

LOC (LA[K]) is address of K^{th} element of LA

W is number of words per memory location for LA

and LB is lower bound i.e. smallest index of LA.

iv) The memory representation of an array is shown in figure below :-



Q. 14 Consider the array AUTO, which records number of automobiles from 1932 thr 1984. Suppose Base (AUTO) = 200 and W = 4 words. Then,

LOC (AUTO [1932]) = 200, LOC (AUTO [1933]) = 204

LOC (AUTO [1934]) = 208

Calculate address at which 1965's record is stored.

Ans. :

Given :-

K = 1965

Base address = 200

W = 4

LB = 1932

The address of the array element for the year 1965 can be obtained –

$$\text{LOC (AUTO [1965])} = \text{Base (AUTO)} + W\,(1965 - LB)$$

$$= 200 + 4\,(1965 - 1932) = 332$$

Q. 15 What is traversing an array ? Give the algorithm for traversing a linear array.

(Oct. 2005,12,15; March 2006,.March 2018; Ju

Ans. :

Traversing an array means accessing with each element of array only at once, so t can be processed.

Algorithm : *Traversing a linear array.*

Here LA is a linear array with lower bound LB and upper bound UB. Following algor apply operation PROCESS to each element of LA.

```
Step 1 :  [Initialize counter]
          set K : = LB
Step 2 :  Repeat steps 3 and 4 while K ≤ UB :
```

```
Step 3 :   [Visit element]
           Apply PROCESS to LA[K]
Step 4 :   [Increment counter]
           set K := K + 1
           [End of step 2 loop]
Step 5 :   Exit
```

OR

This algorithm traverses a linear array LA with lower bound LB and upper bound UB.

```
Step 1 :   Repeat FOR K = LB To UB :
           Apply PROCESS to LA[K]
           [End of loop]
Step 2 :   Exit
```

Q. 16 What is inserting ? Write an algorithm for inserting an element to a linear array. **(Mar. 2009,11,16)**

Ans. : **(Oct.2015)**

i) Inserting refers to the operation of adding an element to the existing elements of array.

ii) The element can be easily inserted at the end of array. But for insertion in the middle of array, it is required to move the elements of array one byte forward.

iii) The following algorithm inserts a data element ITEM into the Kth position in an array LA with N elements.

Algorithm :

INSERT (LA, N, K, ITEM)

Here LA is a linear array with N elements and K is a positive integer, such that K δ N. This algorithm inserts an element ITEM at K^{th} position in LA.

```
Step 1 :   [Initialize counter]
           Set J := N
Step 2 :   Repeat steps 3 and 4 while J ≥ K :
Step 3 :   [Move $J^{th}$ element forward]
           Set LA[J + 1]: = LA[J]
Step 4 :   [Decrement counter]
           Set J := J – 1
           [End of step 2 loop]
Step 5 :   [Insert the element]
           Set LA[K] := ITEM
Step 6 :   [Reset N]
           Set N := N + 1
Step 7 :   Exit
```

Q. 17 What is deleting ? Write an algorithm for deletion of an element from an array. **(Oct. 07**

Ans. :

i) Deleting means removing an element from the existing elements of an array.

ii) Deletion at the end of an array is easier. But, if to delete an element from mid of array, then to move the elements of array one location upward.

iii) **Algorithm :** DELETE (LA, N, K, ITEM)

Here LA is a linear array with N elements and K is a positive integer, such that K = N This algorithm deletes Kth element from LA and assigns it to variable ITEM.

```
Step 1 :  Set ITEM := LA[K]
Step 2 :  Repeat For J = K to N – 1 :
          [Move (J + 1)st element backward]
          Set LA[J] := LA[J + 1]
          [End of loop]
Step 3 :  [Reset number N of elements in LA]
          Set N := N – 1
Step 4 :  Exit
```

Q. 18 Suppose a company keeps a linear array year, such that year [K] contains numbe employees in year K. Write a module for each of the following tasks :

(a) To print each of the years in which no employee was born.

(b) To find the number N of years in which no employee was born.

(c) To find the number NL of employees, who will be atleast L years old at the of year 1984.

Linear array year contain elements from 1920 to 1970.

Ans. :

(a) To print each of the years in which no employee was born.

```
1.  Repeat for K := 1920 to 1970 :
        if year [K] = 0, then :
            write : K
        [End of If structure]
    [End of loop]
2.  Exit
```

(b) To find number N of years, in which no employee was born.

```
1.  Set N := 0
2.  Repeat for K = 1920 To 1970 :
        If year [K] = 0, then :
            N := N + 1
        [End of If structure]
        [End of loop]
3.  Write : N
4.  Exit
```

(c) To find number of employees NL, who will be at least L years old at the end of year 1984 we want the number of employees born in year 1984-L or earlier.

```
1.   Set NL := 0
2.   Set X := 1984 - L
3.   Repeat For K = 1920 To X :
     Set NL := NL + year [K]
     [End of loop]
4.   Write : NL
5.   Exit
```

Q. 19 Explain Bubble sort algorithm with suitable example.

(March 2002, 05, 08, 12, 17, 20; Oct. 2005, 2008)

Ans. : Algorithm :

Bubble Sort (DATA, N)

Here DATA is a linear array with N elements. This algorithm sorts elements of DATA in ascending order.

```
Step 1 : Repeat steps 2 and 3 for K := 1 To N - 1 :
Step 2 : Set Ptr := 1
Step 3 : Repeat While Ptr ≤ N - K :
         (a) If DATA [Ptr] > DATA [Ptr + 1], then interchange
         DATA [Ptr] and DATA [Ptr + 1]
             [End of If structure]
         (b) [increment pointer]
             Set ptr := ptr + 1
             [End of inner loop]
             [End of outer loop]
Step 4 : Exit
```

Explanation :

Suppose DATA is an array of N elements. Sorting these elements in ascending order means arranging the elements such that :

DATA [1] < = DATA [2] < = ... < = DATA [N]

In Bubble sort, compare DATA[1] with DATA[2] and exchange them if DATA[1] > DATA[2].

Next DATA[2] is compare with DATA[3]. They are exchanged if necessary. This process is repeated till DATA[N – 1] is compared with DATA[N].

One makes N – 1 comparisons, this is called a **pass.**

After the first pass the largest element is sink to the last position.

During the next pass, compare elements upto the last but one and second largest element moves to the (N – 1)st position.

After N – 1 passes, all elements are sorted.

Consider a linear array consisting of 5 elements, given below :

Data[1]	55
Data[2]	43
Data[3]	05
Data[4]	06
Data[5]	09

Pass 1 :

(a) Compare DATA[1] with DATA[2] since 55 > 43 ∴ exchanged

55 43 5 6 9

∴ New list is 43 55 5 6 9

(b) Next compare DATA[2] with DATA[3] since 55 > 5 ∴ exchanged

43 55 5 6 9

∴ New list is 43 5 55 6 9

(c) Now, compare DATA[3] with DATA[4]since 55 > 6 ∴ exchanged

43 5 55 6 9

∴ New list is 43 5 6 55 9

(d) Compare DATA[4] with DATA[5] since 55 > 9 ∴ exchanged

43 5 6 55 9

∴ New list is 43 5 6 9 55

At the end of first pass, the largest element 55, has moved to the last position.

Pass 2 : In this pass, only three comparisons since K = 2.

(a) 43 5 6 9 55 Since 43 > 5 ∴ exchanged

∴ New list is 5 43 6 9 55

(b) 5 43 6 9 55 Since 43 > 6 ∴ exchanged

∴ New list is 5 6 43 9 55

(c) 5 6 43 9 55 Since 43 > 9 ∴ exchanged

∴ New list is 5 6 9 43 55

At the end of second pass, the second largest element 43 has maked to its p position.

Pass 3 :

5 6 9 43 55 Since 5 < 6 ∴ No exchange

∴ 5 6 9 43 55 Since 6 < 9 ∴ No exchange

∴ New list is 5 6 9 43 55

Pass 4 :

In this way after complete execution of this algorithm, the array gets sorted in ascending order as follows :

DATA[1]	05
DATA[2]	06
DATA[3]	09
DATA[4]	43
DATA[5]	55

Q. 20 What do you understand by the term searching ? Which are the different types of searching algorithms ? Explain the linear searching algorithm.

(March 2004, Oct. 2004, 2010)

Ans. : Searching : Searching means to find out particular element from a given list of elements or check whether required element is present or not in an array. There are two types of searching algorithms as follows :

(1) Linear search (2) Binary search

Linear searching algorithm :

In linear search the given element is compared with each element of list one by one. For algorithm, refer to Q. No. 21.

Q. 21 Write an algorithm for linear search technique with suitable example.

(March 2003, 2007, 2009; Oct. 2007, 2010; July 2016)

Ans. :

Algorithm : Linear Search

LINEAR(DATA, N, ITEM, LOC)

Here DATA is a linear array with N elements and ITEM is given element. This algorithm finds the location LOC of ITEM in DATA or sets LOC = 0, if search is unsuccessful.

```
Step 1 :   [Insert ITEM at the end of DATA]
           Set DATA [N + 1] := ITEM
Step 2 :   [Initialize counter]
           Set LOC := 1
Step 3 :   [Search for item]
           Repeat While DATA [LOC] ≠ ITEM :
           Set LOC := LOC + 1
           [End of loop]
Step 4 :   If LOC = N + 1, then :
           Set LOC := 0
Step 5 :   Exit
```

For example : Given DATA array with following 5 elements

11 22 33 44 55

Suppose ITEM = 33

Step 1 : Set DATA [6] = 33, List becomes

11 22 33 44 55 33

Step 2 : LOC = 1

Step 3 : Since DATA [1] = 11 ≠ 33 ∴ LOC = 2

Since DATA [2] = 22 ≠ 33 ∴ LOC = 3

Here DATA [3] = 33 = 33 = ITEM

Step 4 : Hence ITEM = 33 found at position, LOC = 3.

Q. 22 Write an algorithm for binary search technique with example. (Oct. 2002,06,11,12,1

Ans. : Binary search is used to search an element from sorted array. (Mar.2013, 14, 15,1

Algorithm : Binary search

Binary (DATA, LB, UB, ITEM, LOC)

Here DATA is a sorted array with lower bound LB and upper bound UB. ITEM is gi element. BEG denotes beginning, MID denoted middle and END denotes end loca of DATA. This algorithm finds the location LOC of ITEM in DATA or sets LO NULL, if search is unsuccessful.

```
Step 1 :  [Initialize Variables]
          Set BEG : = LB, END := UB and MID := INT ((BEG + END)/2)
Step 2 :  Repeat steps 3 and 4
          while BEG = END AND DATA[MID] ITEM
Step 3 :  If ITEM < DATA[MID], then :
          set END := MID - 1
          Else :
          Set BEG := MID + 1
          [End of If structure]
Step 4 :  Set MID := INT ((BEG + END)/2)
          [End of step 2 loop]
Step 5 :  If DATA[MID] = ITEM, then :
          set LOC := MID
          Else :
          LOC := NULL
          [End of If structure]
Step 6 :  Exit
```

e.g. Given DATA be the following sorted 13 element array :

11 22 30 33 40 44 55

60 66 77 80 88 99

Suppose ITEM = 40

Step 1 : Initially BEG = 1 and END = 13

Hence MID = INT[(1 + 13)/2] = 7

and so DATA[MID] = DATA [7] = 55

Step 2 : Since 40 < 55, END has its value changed by

END = MID – 1 = 7 – 1 = 6

Hence MID = INT [(1 + 6)/2] = 3

and so DATA[MID] = DATA[3] = 30

Step 3 : Since 40 > 30, BEG has its value changed by

BEG = MID + 1 = 3 + 1 = 4

Hence MID = INT [(4 + 6)/2] = 5

and so DATA[MID] = DATA[5] = 40

∴ Found ITEM in location LOC = MID = 5

Q. 23 Explain the advantages of binary search algorithm with a suitable example. State any two disadvantages or limitations of binary search. **(March 07, 19; Oct. 03)**

Ans. :

Advantages of binary search algorithm :

(1) Binary search algorithm is efficient as the search scope gets reduced to half the size of the array, with each iteration.

(2) The number of comparisons required are approximately equal to $\log_2 n$ which are less than linear search.

(3) For example :

Given array data with 7-sorted elements :

11 22 30 33 40 44 55

Suppose ITEM = 40

Step I : Initially BEG = 1 and END = 7

∴ MID = (BEG + END) / 2 = (1 + 7) / 2 = 4

∴ DATA [MID] = DATA [4] = 33

Step II : Since 33 < 40, BEG is changed as BEG = MID + 1 = 4 + 1 = 5

∴ MID = (5 + 7) / 2 = 6

∴ DATA [MID] = DATA [6] = 44

Step III : Since 44 > 40, END has its value changed by END = MID – 1 = 6 – 1 = 5

∴ MID = (5 + 5) / 2 = 5

∴ DATA [MID] = DATA [5] = 40

∴ ITEM found at location 5 in array.

In above example, only two comparisons are required because at each iteration MID is calculated only one half is checked.

In the same example, for linear search, 5 comparison are required.

Disadvantages :

1) The given list must be sorted.

2) The access of list must be random means the middle element can be accessed.

3) At each iteration, middle entry calculation is required.

Q. 24 Write difference between Linear search and Binary search. **(Mar.2014, 2017, July**

Ans. :

Linear Search Binary Search		Binary Search	
1.	Linear search performs on unsorted list of elements as well as sorted list.	1.	For binary search, the elements array are stored in alphabetically numerically in sorted manner.
2.	Compare the desired element with all elements in an array until the match is found	2.	Compare the value of midpoint wi desired value. If the value is grea than midpoint value, the first half checked, otherwise second half checked until search is successful interval is empty.
3.	Insertion of an element in an array can be performed very efficiently when array is not ordered.	3.	An insertion of a new elemen requires that many elements b physically moved to preserved order
4.	For large size of array, time required for this search is very large.	4.	For large size of array, comparativel time required is less.
5.	Time complexity is as follows : worst case : N comparison Best case : 1 comparison	5.	Time complexity as follows : worst case : $\log_2 N$ comparison Best case : 1 comparison

Q. 25 What are pointer arrays ? **(Oct. 2003,06; Mar. 2012,15,19, July 20**

Ans. :

i) An array is called pointer array, if each element of that array is a pointer.

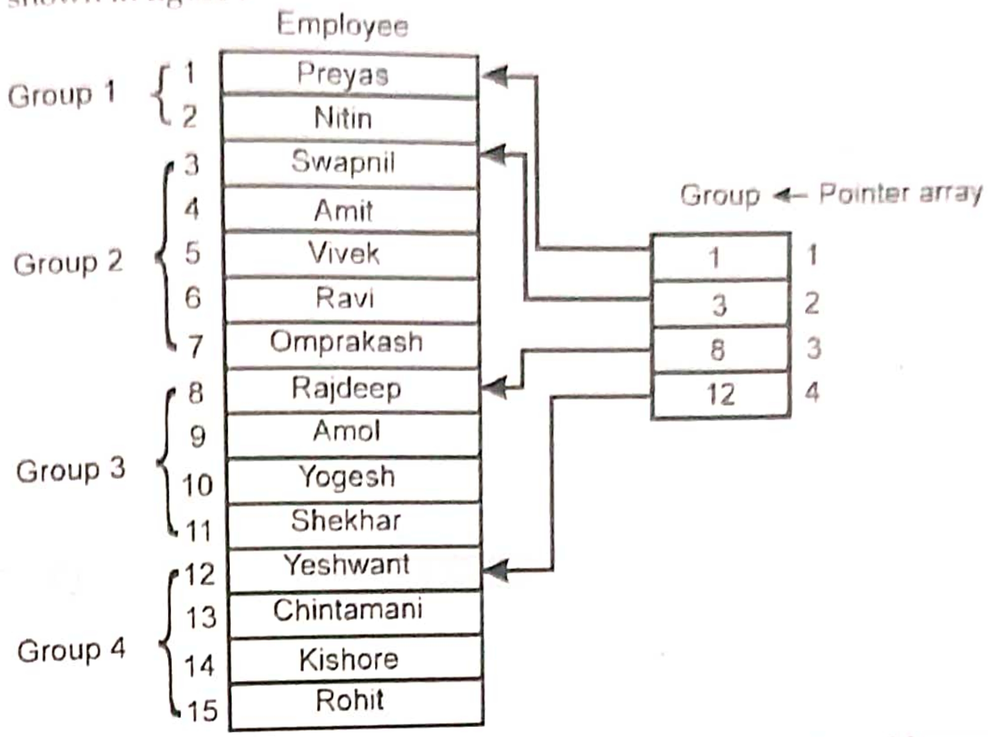
ii) The variable is called as pointer variable, if it points to another variable i.e. it conta the memory address of other variable.

iii) Consider an organization, which divides its employee list into four groups, depend on certain conditions. Following figure shows the list of 4 groups. There are employes and groups contain 2, 5, 4 and 4 employes respectively as

Group 1	Group 2	Group 3	Group 4
Deepak	Swapnil	Rajdeep	Yashwant
Nitin	Amit	Amol	Chintamani
–	Vivek	Yogesh	Kishore
–	Ravi	Shekhar	Rohit
–	Omprakash	–	–

iv) If these groups are to be represented in memory, the most efficient way is to 2 arrays. The first is Employee array, which contains list of employees in all four gro sequentially, while the second array is Group array, which is a pointer array, wh contains the starting address of each group in the Employee array, respectively.

It is shown in figure :



i) Each element of Group array is a pointer, which holds the starting addresses of different groups. Hence, it is called as pointer array.

. 26 **What is a record ?** **(March 2011, 18,19)**

ns. :

) A record is a collection of relative data items, each of which is called as field or attribute.

) Collection of records is known as files. Collection of data is frequently organized into a hierarchy of fields, records and files.

i) A record may contain non-homogeneous data i.e. data items of record need not to be of same data type. In a record, natural ordering of elements is not possible. The elements in record can be described by level number.

) e.g. An organization keeps records of its Employees. It contains following data items- Name, Sex, Salary, Birthday, Address.

Name is group item consisting of First name, Middle name and Last name. Also, Birth date and Address are group items.

The structure of this record is shown in figure below.

1. Employee
    2. Name
        3. First name
        3. Middle name
        3. Last name
    2. Sex
    2. Salary
    2. Birth date
        3. Date

3. Month

3. Year

2. Address

3. City

3. Pincode

v) The number to the left of each variable indicates level number.

vi) Employee (30)

This indicates a file of 30 records.

vii) To access first name of 3rd employee, we should write Employee (3).Name.First name

In this way, we can access variables in records.

Q. 27 What is a record ? How it differs from a linear array ?

(March 2002; 05, 07, 08,14,16;Oct. 10,11; July

Ans. : A record is a collection of fields or attributes i.e. relative data items. Collection of dat frequently organized into hierarchy of fields i.e. records. A file is nothing but collect of records.

Difference between records and linear arrays :

(i) A record is a collection of fields, while an array is list of homogeneous data elements.

(ii) A record may contain non-homogeneous data i.e. data elements may be of different types. An array always contains homogeneous data.

(iii) In a record, natural ordering of elements is not possible. Array elements can be natu ordered.

(iv) Elements of record are referenced by level number, while those of array car referenced by an index set consisting of n consecutive numbers.

Q. 28 How records are represented in memory using array ?

(Oct. 2002; March 2004, March 2

Ans. :

i) Consider a record, whose structure is given below.

1. Employee

2. Name

3. First name

3. Last name

2. Sex

2. Address

3. City

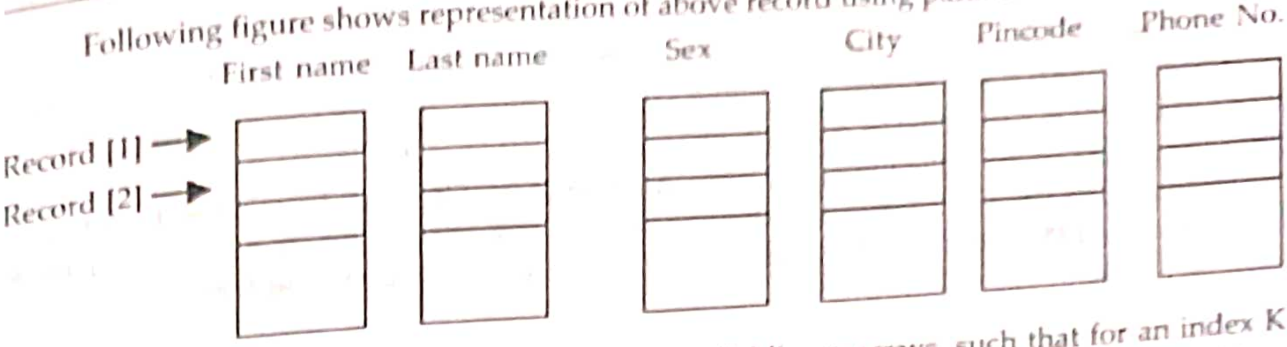
3. Pincode

2. Phone no.

ii) To represent this record in memory, linear arrays are used.

iii) One separate linear array is used for each elementary item of record such as First n Last name, Sex, City, Pincode, Phone no.

Following figure shows representation of above record using parallel linear arrays.



iv) The records are stored in memory using parallel linear arrays, such that for an index K of all records, First name [K], Last name [K], Sex [K], ... belong to the same record in a file. (i.e. Kth record in the file)

Q. 29 Show representation of records in memory considering suitable example of three records and three fields. **(Mar. 2003,11,13; Oct.2011,13)**

Ans. :

1) Records contain non-homogeneous data, so it cannot be stored in array.

2) But in entire file of records, all data elements belonging to the same identifier will be of same type. So a file may be stored in memory as collection of arrays.

3) One array for each of data item. All the arrays should be parallel.

4) For e.g.

A student file consisting three records and three fields.

Name	Address	Phone
Lokesh	11, J.M. Road	5662000
Jayesh	24, M.G. Road	4240020
Anushka	10, Sahkarnagar	4261900

Following figure shows representation of above file in three parallel arrays Name, Address and Phone –

Name
Lokesh
Jayesh
Anushka

Address
11, J.M. Road
24, M.G. Road
10, Sahkarnagar

Phone
5662000
4240020
4261900

All arrays should be parallel that is for subscript K the elements Name [K], Address [K], Phone [K] must belong to same record.

Linked List

Q. 30 What are linked lists ? Show a liked list with suitable example having six nodes with a properly labelled diagram. **(Mar. 2002,04,05,06,07,08,13,14,15; Oct. 2003,07,14)**

OR

With suitable example, show labelled diagram for link between two nodes having the information part and next pointer field.

What are linked lists ? Show a liked list with suitable example having five nodes with a properly labelled diagram. **(Mar. 2013, March 2020)**

Ans. :

i) A linked list is a linear collection of data elements, called nodes, where the linear o is maintained with the help of pointers.

ii) Linked list is also called as one-way list.

iii) Each node in the linked list is divided into two parts. First part is called as **INFO** p which contains the information about the element or actual element and second pa called as **LINK** part, which is next pointers field i.e. it contains the address of next n in the list.

iv) e.g.

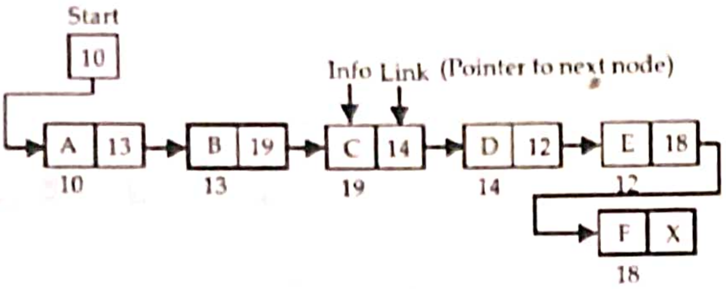


Fig. 1 : Linked list with 6 nodes

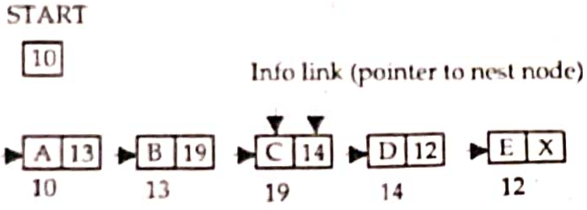


Fig. 2 : Linked list with 5 nodes

(a) The left part of the node is the Info part, which contains information of the eleme while the right part is Link part, which is next pointers field i.e. it points to next node.

(b) An arrow is drawn from Link part of one node to the next node to indicate link.

(c) The address of the list is stored in Start or Name of the list.

(d) The Link part of last node is NULL pointer i.e. it contains nothing.

(e) To trace the linked list, we just require the address of Start or Name.

Q. 31 What are the advantages of linked lists over linear arrays ? **(March 20**

Ans. : Advantages of linked lists over arrays :

(i) To store arrays in memory, require consecutive memory locations, while to store lin lists, consecutive memory locations are not required.

(ii) Arrays can not be easily extended, while linked list can be easily extended.

(iii) There is very complicated procedure to insert an element in an array. One can ea insert an element in an linked list.

(iv) Similarly, deletion of an element from array is very complicated, while deletion fr linked list is easy.

(v) Linked lists can be easily implemented and maintained in computer memory.

32 How linked lists are represented in memory ? OR

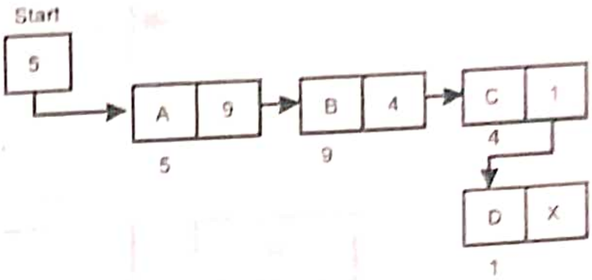
(March 2003,12,14, 17, 19; Oct. 2006, 07,13, July 16, 18)

With suitable example show the representation of linked list in memory.

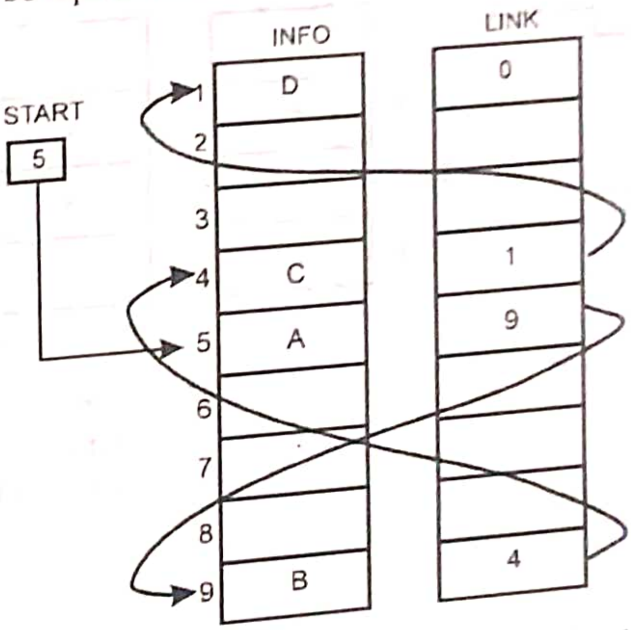
s. :

Linked lists can be represented in memory by using two arrays respectively known as INFO and LINK, such that INFO[K] and LINK[K] contains information of element and next node address respectively.

The list also requires a variable 'Name' or 'Start', which contains address of first node. Pointer field of last node denoted by NULL which indicates the end of list. e.g. Consider a linked list given below :



The linked list can be represented in memory as –



Above figure shows linked list. It indicates that the node of a list need not occupy adjacent elements in the array INFO and LINK.

. 33 Explain insertion and deletion from linked list with example.

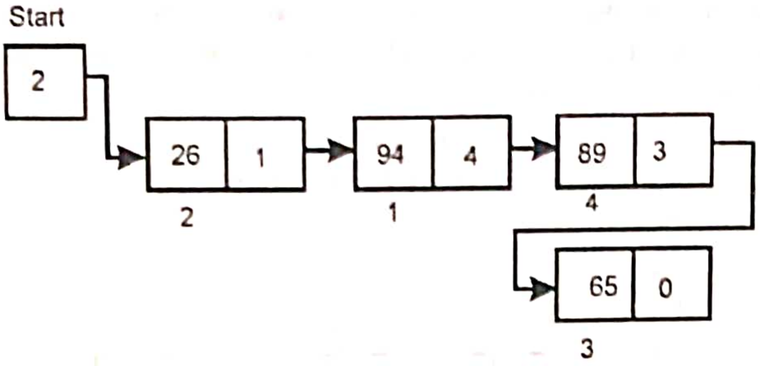
ns. :

It is easier to insert an element into or delete an element from a linked list than arrays.

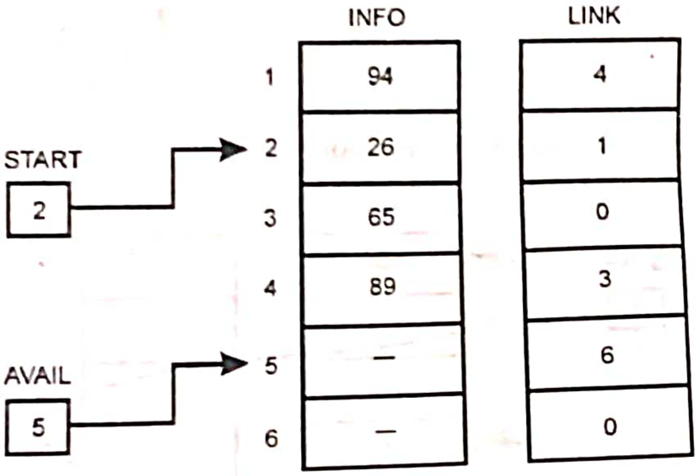
(i) **Insertion into a linked list :**

For insertion of an element into a linked list, the only requirement is that free mem space is available to store a node.

e.g. Consider a linked list having four nodes as follows.



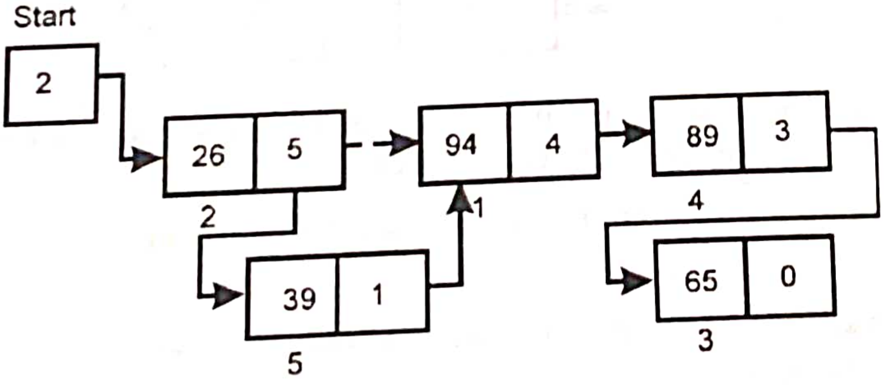
This list can be represented in memory as :



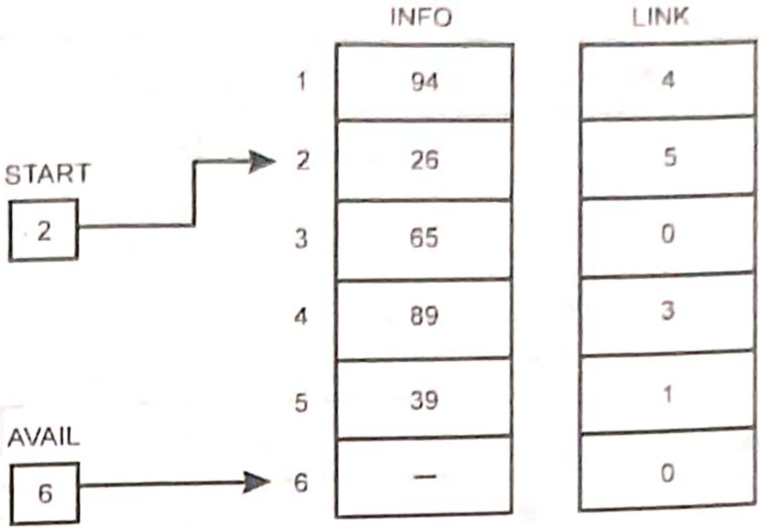
	INFO	LINK
1	94	4
2	26	1
3	65	0
4	89	3
5	—	6
6	—	0

Now, to insert an element on second position of the list, the content of AVAIL are sto in LINK part of first node (since, AVAIL points to the memory location where new no can be inserted) and LINK part of the first node is transferred to LINK part of new no

Then the list can be represented as follows.



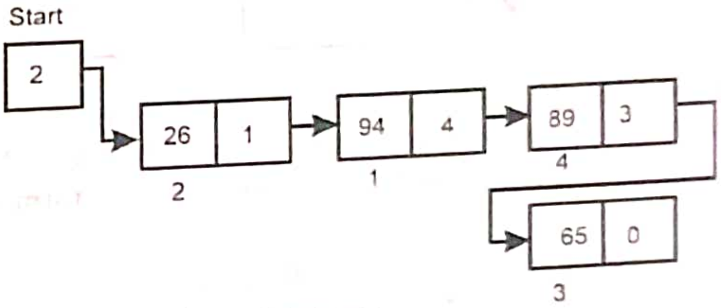
This list can be represented in memory as :



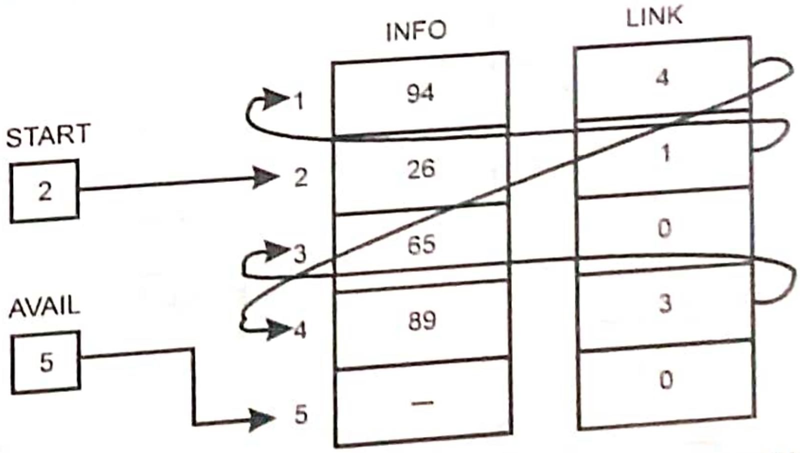
(ii) Deletion from linked list :

To delete a node from a linked list, the LINK part of that node is given to the LINK part of the previous node.

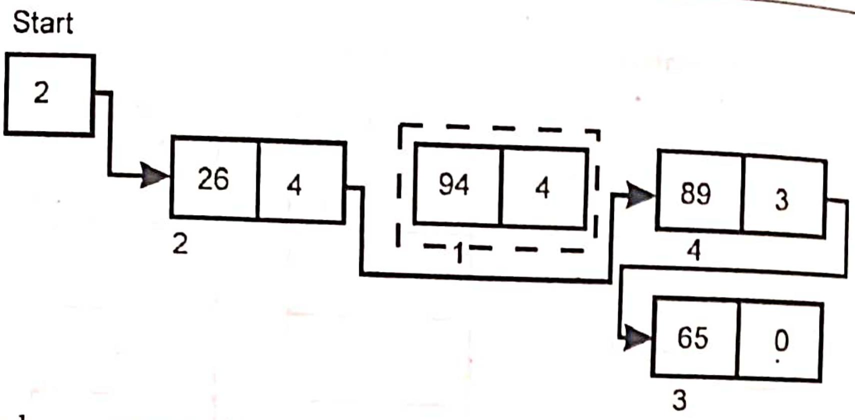
e.g. Consider a linked list as follows :



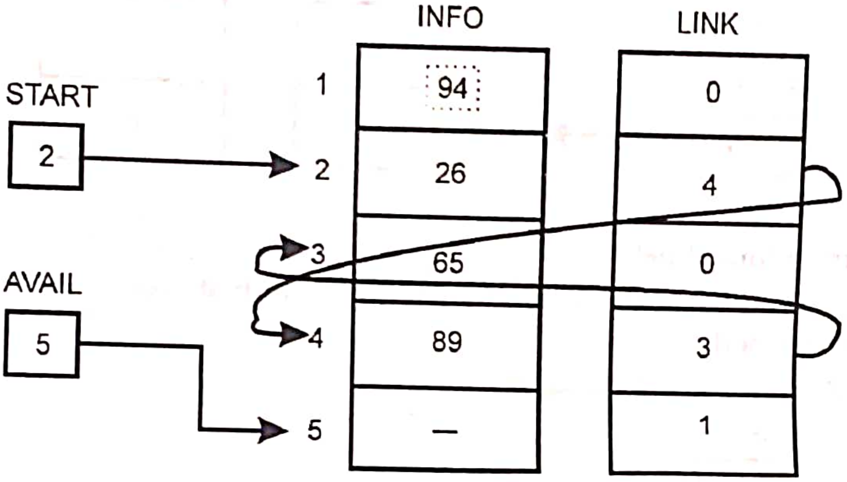
This list can be represented in memory as :



Now, to delete second node from the list, then just transfer the LINK part of second node to the LINK part of the first node.



This list can be represented in memory as



Q. 34 **There is a list of 5 hospital patients and their corresponding room numbers. Fill th values of N start and N link so that they form an alphabetical link of patient names Also fill the values of R start and R link so that they form an ordering of room numbers.**

N start □

R start □

	Name	Room No.	N link	R link
1	Deepak	127		
2	Nitin	021		
3	Swapnil	420		
4	Amit	040		
5	Vivek	199		

Ans. :

N start 4

R start 2

	Name	Room No.	N link	R link
1	Deepak	127	3	5
2	Nitin	021	1	4
3	Swapnil	420	5	0
4	Amit	040	2	1
5	Vivek	199	0	3

Q. 35 **The following figure pictures a linked list in memory.**

Start

4

Avail

3

	INFO	LINK
1	A	2
2	B	8
3		6
4	C	7
5	D	0
6		0
7	E	1
8	F	5

(i) Find the sequence of characters in the list.

(ii) Suppose F and then C are deleted from the list. After that G is inserted at the beginning of the list. Find the final structure of the list.

(iii) Suppose G is inserted at the beginning of the list and then F after that C is deleted from the list. Find the final structure of the list.

Ans. :

i) Linear order of characters

START = 4 so INFO [4] = C is the first character

LINK [4] = 7 so INFO [7] = E is the second character

LINK [7] = 1 so INFO [1] = A is the third character

:

:

LINK [5] = 0, the NULL value so the list is ended.

∴ C E A B F D is the character string. Hence sequence is C, E, A, B, F, D.

(ii)

START

3

AVAIL

6

	INFO	LINK
1	A	2
2	B	5
3	G	7
4	C	0
5	D	0
6	–	8
7	E	1
8	F	4

Sequence

G, E, A, B, D

(iii)

START: 3

AVAIL: 6

	INFO	LINK
1	A	2
2	B	5
3	G	7
4	Ⓒ	0
5	D	0
6		8
7	E	1
8	Ⓕ	4

Sequence

G, E, A, B, D

Q. 36 **Let LIST be a linked list in memory. Write an algorithm for traversing the link list for following purposes :**

(i) Find the number of times given ITEM occurs in the list.

(ii) Find number of non-zero elements in the list.

(iii) Add given value K to each element of the list.

Ans. : Algorithm : Traversing a linked list

```
1.  Set Ptr := START
2.  Repeat While Ptr ≠ NULL :
        Apply process to INFO[Ptr]
        Set ptr := LINK [ptr]
        [End of loop]
3.  Exit
```

(i)

```
1. Set Ptr := START
2. Set N := 0
3. Repeat steps 4 and 5 While Ptr NULL :
4.    If INFO [ptr] = ITEM, Then :
      set N := N + 1
      [End of If structure]
5. Set Ptr := LINK [ptr]
     [End of step 3 loop]
6. Write : N
7. Exit
```

(ii)

```
1. Set Ptr := START
2. Set N := 0
3. Repeat steps 4 and 5 While Ptr ≠ NULL :
4. If INFO [Ptr] ≠ 0, Then :
     Set N := N + 1
     [End of If structure]
```

```
      5. Set Ptr := LINK [Ptr]
         [End of step 3 loop]
      6. Write : N
      7. Exit
(iii) 1. Set Ptr := START
      2. Repeat While Ptr ≠ NULL :
           Set INFO [Ptr] := INFO [Ptr] + K
           Set Ptr := LINK [Ptr]
           [End of loop]
      3. Exit
```

Stack and Queue

Q. 37 Explain Stack and Queue with suitable examples. OR **(Mar.2013, July 2017)**

Explain LIFO and FIFO Systems with suitable examples. **(Oct. 2005, 2010)**

Ans. : LIFO System :

(i) LIFO system is last-in-first-out system. In this type of system, the element which is inserted at last, will be deleted first.

(ii) Stack is an example of LIFO system. It is a linear system in which insertion and deletion takes place only at one end i.e. top of the list.

(iii) The insertion operation is referred to as **push** and deletion operation as **pop**.
e.g. consider a stack of dishes. If we want to add a new dish to this stack then it is added at the top of stack also deletion takes place from the top.

FIFO System :

(i) A FIFO system is first-in-first-out system. In this type of system, the element which is inserted first in the list will also be deleted first.

(ii) Queue is an example of FIFO system. A queue is a linear list, in which insertion takes place only at one end of the list known as 'rear' of the list and deletion takes place at the other end, called as 'front' of the list.
e.g. A queue for tickets in a cinema hall.

Tree

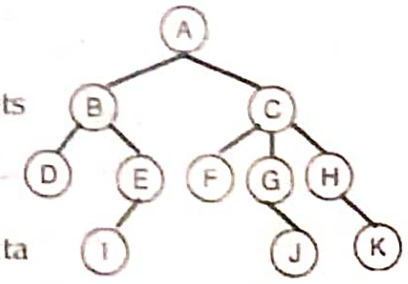
Q. 38 What is a tree ? What do you mean by root, leaf, siblings and child about tree. **(Oct. 2006, 2010)**

Ans. : Tree :

Tree is a non-linear hierarchical data structure which consists of finite set of one or more nodes (i.e. collected data items) such that :

a) There is specially designated node called the root.

b) The remaining nodes are partitioned into n≥0 finite disjoint sets T1, T2,...., Tn where each of these set is tree.
T1, T2, ..., Tn are called 'subtrees' of the root.
For e.g. figure shows tree which has 11 nodes, each item of data being a single letter.



Root :

A node which has no parent. Generally first node is called as 'root node'. In figure, a th node A is the root of the tree.

Leaf : (July 201

The node which has no child or children. Such nodes have degree zero. In figure a D, F, J, K are the leaf nodes. Also called as terminal node.

Child :

The nodes which are reachable from a node, say u, through a single edge are called th children of u. e.g In figure a, the children of node C are F, G, and H.

Sibling : (July 201

Children of the same parent are said to be siblings. e.g. The nodes D and E are bot children of node B. So D and E are siblings.

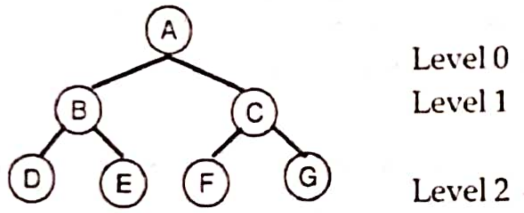
Q. 39 Explain the following terms :

Ans. : 1. Level of tree :

Each node in a tree is assigned a level number. Generally, the level number of root 'R' o the tree is zero and every other node is assigned to level number which is one more thar the level number of its parent.

It is the distance from the root.

For e.g.



2. Depth / Height : (March 2017

Depth of a tree is defined as maximum level of any node in a tree. If root is level 0 ther depth or height of tree is equal to 1 + largest level number.

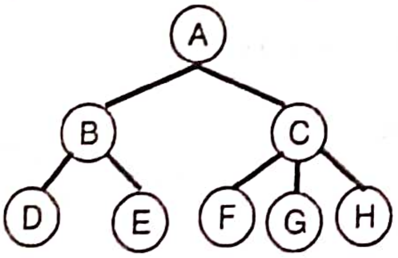
e.g. Depth of above tree is 3.

3. Degree :

The number of subtrees of a node is called degree of a node.

The degree of a tree is the maximum degree of the node in tree.

e.g. the degree of each node in figure is as



Node	Degree
A	2
B	2
C	3
D, E, F, G, H	0

The tree has degree 3.

. 40 What is a binary tree ? **(March 2002,04,05,14,15,19, 20; Oct. 2004,06,11,12,13)**

ns. : Binary tree is a finite set of elements called nodes such that is:

It may be empty or

It is partitioned into three disjoint subsets :

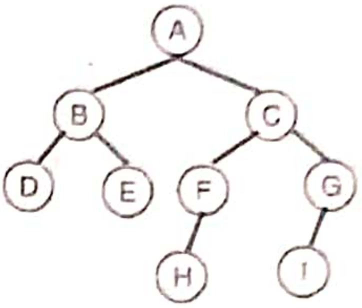
) there is a single distinguished element called the root of tree.

) other two subsets are themselves binary tree called left subtree and right subtree of the original tree.

A left and right subtree can be empty.

In binary tree, there is no node with degree greater than two.

e.g.

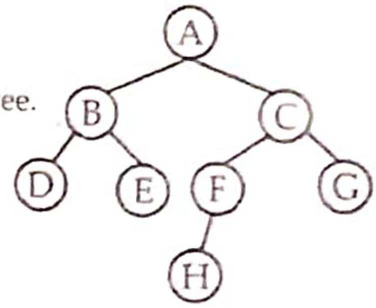


. 41 What is a binary tree ? With a suitable example, explain the terminology describing family relationship between the elements of a binary tree. **(March 2005, 11; July 18)**

ns. : Binary tree : Please refer Q. No. 40.

Basic terminology : Consider the example :

The binary tree contains 8 nodes (A to H). Root A is at the top of tree.



) Left successor : B is left successor of node A.

) Right successor : C is right successor of node A.

) Left subtree : Left subtree consists of nodes B, D and E.

) Right subtree : Right subtree consists of nodes C, F, G and H.

) Terminal node : The node with no successors are is called terminal node D. E, H and G are terminal nodes.

) Binary tree T1 and T2 are similar if they have same structure.

Any node N in a binary tree T has either 0, 1 or 2 successors.

. 42 What is a binary tree ? With a suitable example show the relationship between total numbers of nodes and depth of a tree. **(Oct. 2003,15, March 2006, March 2018)**

ns. :

inary tree : Please refer Q. No. 40.

elationship between total number of nodes and depth of a tree :

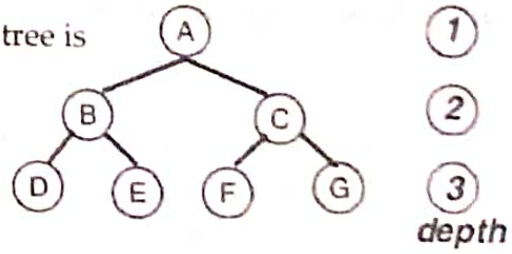
Depth of a tree means maximum level of any node in a tree. Maximum number of nodes of binary tree with depth n are $2^n - 1$.

For example :

Consider the following tree with depth 3.

So with depth 3, the total number of nodes in a given tree is

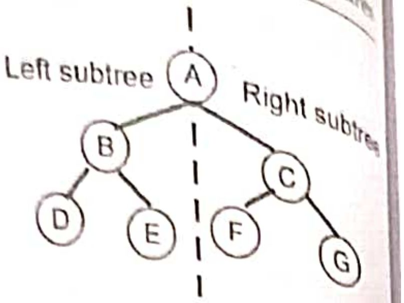
$$\begin{aligned} 2^n - 1 &= 2^3 - 1 \\ &= 8 - 1 \\ &= 7 \end{aligned}$$



The tree with depth n having $2^n - 1$ number of total nodes.

Q. 43 Define the following :

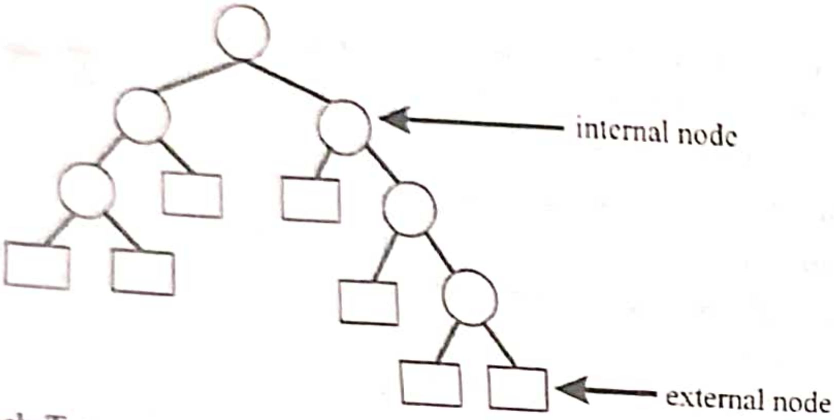
Ans. :



1. **Complete Binary Tree :** If all leaf nodes of a binary tree have same level number and every non-leaf node has non-empty left and right subtrees then the tree is called as complete binary tree. All nodes at the last level appears as far left as possible.

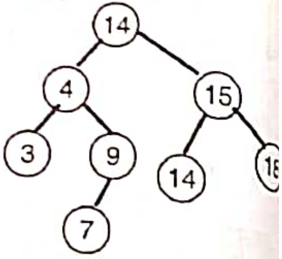
2. **Extended binary tree or 2-tree :** **(Mar.2015**

A binary tree T is said to be a 2-tree or an extended binary tree if each node N has either 0 or 2 children. The nodes with 2 children are called internal nodes and the nodes with children are called external nodes.



3. **Binary Search Tree :**

It is a binary tree in which each node N of tree has the property that the value at N is greater than every node value in the left subtree of N and is less than or equal to every node value in the right subtree of N.



Q. 44 How binary trees are represented in memory ? OR **(Mar.2015, 20, July 2016, 19**

With suitable example and labelled diagram, show the representation of binary tree in memory. **(March 2003, 2009**

Ans. :

A binary tree T can be represented in memory by two types of representation :

(i) Linked representation

(ii) Sequential representation.

(i) **Linked representation :** **(Oct. 2008,15**

Linked representation uses three parallel arrays INFO and, LEFT and RIGHT and a pointer variable ROOT such that for an index K, INFO [K] contains actual element, LEFT [K] contains address of left child and RIGHT [K] contains address of right child.

e.g. Consider a binary tree as below :

can be represented in memory as,

ROOT

5

AVAIL

10

	INFO	LEFT	RIGHT
1	B	7	14
2	G	12	13
3		0	
4	E	11	0
5	A	1	9
6		3	
7	D	0	0
8		6	
9	C	14	2
10			
11	H	0	0
12	I	0	0
13	J	0	0
14	F	0	0

The ROOT stores address of first node of tree T.

AVAIL stores address of first null node. To insert another node to tree T, it is inserted at memory location pointed by AVAIL.

Note : In above example, to insert an element K, then it will be inserted at INFO [10]. After insertion, LEFT [10] and RIGHT [10] will contain zero (null pointer) and AVAIL will contain 8 i.e. next element is to be inserted at 8.

(ii) Sequential representation :

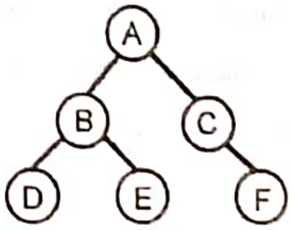
For sequential representation, only one linear array is used. This array is generally known as TREE such that :

(a) The root R of the tree is stored in TREE [1].

(b) If a node N of tree stored in TREE [K], then,

its left successor is stored in TREE [2*K] and right successor is stored in TREE [(2*K) + 1]

e.g. Consider a binary tree as follows :



This tree can be represented in memory as,

	TREE
1	A
2	B
3	C
4	D
5	E
6	–
7	F
8	–
9	–
10	–

In general, the sequential representation of a tree with depth d will require an array with approximately 2^{d+1} elements.

Q. 45 Each store in a chain sends in a weekly record of its sales according to the following structure 01 Store, 02 Hardware, 02 Clothing, 03 Men, 03 Women, 03 Children 04 Boys, 04 Girls, 02 Drugs, 03 Prescription, 03 Nonprescription, 02 Stationary.

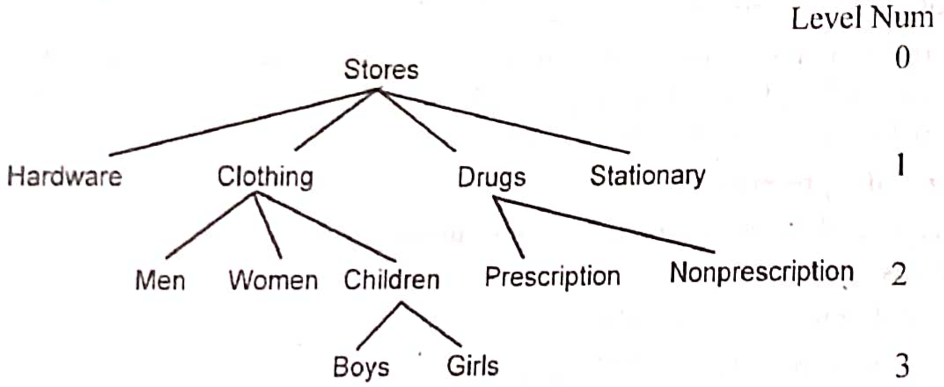
1. Draw the appropriate tree diagram.

2. How many elementary items are there ?

3. How many group items are there ?

Ans. :

1. The tree diagram is as follows :



2. Elementary items are those all nodes which have no children under given group. The above tree has 8 elementary items as :

Hardware, Men, Women, Boys, Girls, Prescription, Nonprescription, Stationary.

3. Group items are those all nodes having children excluding root.

The above tree has 3 group items as -

Clothing

Children

Drugs.

Q. 46 Explain the following data structures with suitable diagram.

(a) Linear array (b) Linked list

(c) Tree **(March 2003,08,11,12,15,19; Oct. 2006)**

Ans. :

(a) **Linear array :**

A linear array is the data structure which consists of finite ordered set of homogeneous data elements such that :

(i) The elements of the array are referenced respectively by an index set consisting on consecutive numbers.

(ii) The elements of the array are stored respectively in successive memory locations.

(iii) The number n of the elements is called length of size or array.

For e.g. let DATA be 5 elements linear array as follows :

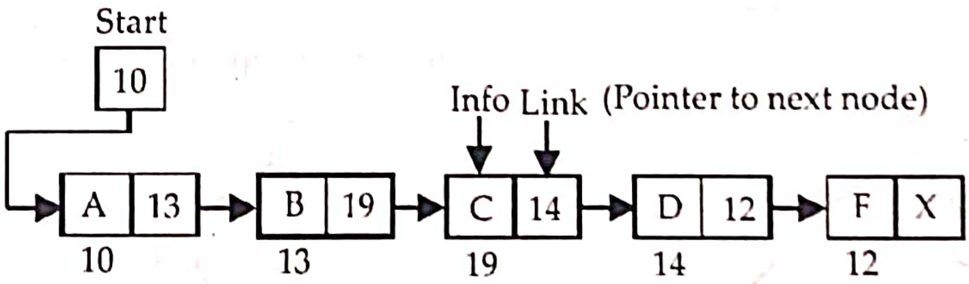
	DATA
1	300
2	400
3	100
4	50
5	09

(b) **Linked list :** **(March 2019)**

(i) A linked list is a linear collection of data elements called nodes where the linear order is maintained with the help of pointers.

(ii) Each node in the linked list is divided into two parts. First part is called as INFO part which contains the information about the element or actual element and second part is called as LINK part which is next pointer field i.e. it contains the address of next node in the list.

(iii)



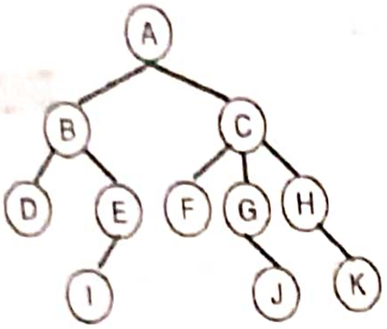
(c) **Tree :** **(Oct. 2006)**

Tree is a non-linear hierarchical data structure which consists of finite set of one or more nodes (i.e. collected data items such that :

(i) There is specially designated node called the root.

(ii) The remaining nodes are partitioned into n Z 0 finite disjoint sets T1, T2, ..., Tn where each of these set is tree.

T1, T2, ..., Tn are called 'subtree' of the root.

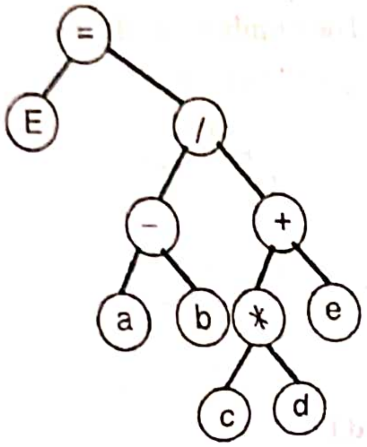


Q. 47 What are binary trees ? Draw the binary tree structure for the following expression :
E = (a + b) / [(c * d) – e] **(March 2004)**

Ans. :

Binary Tree : Refer to Q. No. 40.

The binary tree structure for the expression
E = (a + b) / [(c * d) – e]



Q. 48 Draw the tree structure for the following expressions. **(March 2002,07,08; Oct. 2004,11)**

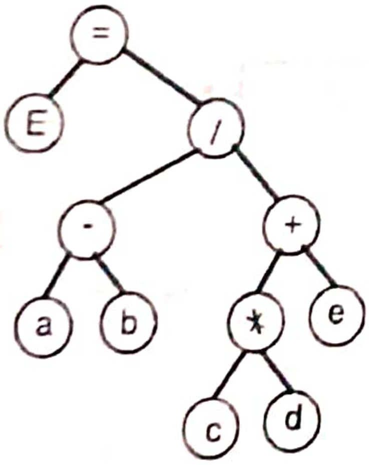
1) (i) E = (a – b) / (c * d) + e) (ii) E = (p – q) / (r * s) + t

2) [(a + b) * c]/[a * ((b – c) + a)]

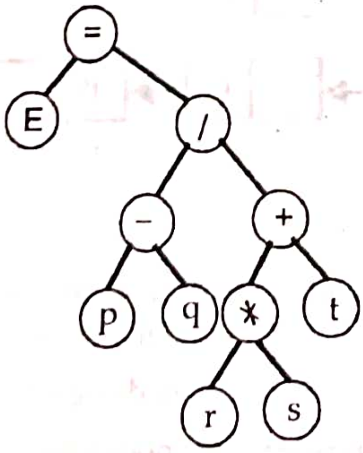
3) (2x + y) (a - 7b)^3

Ans. : 1. E = (a - b)/(c* d) + e)

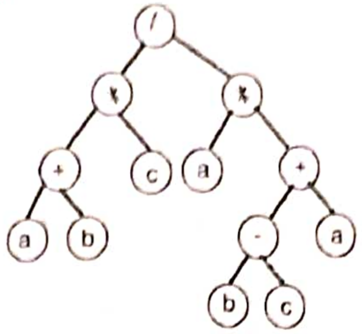
(i)



(ii) E = (p – q) / ((r * s) + t)

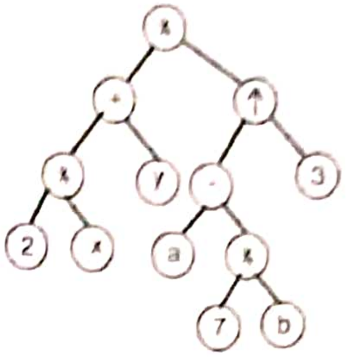


2. [(a + b) * c]/[a * ((b – c) + a)]



3. (2x + y) (a – 7b)^3

(March 2008)

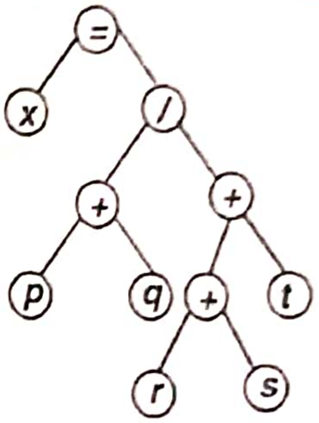


Q. 49 **Draw a tree structure for the following expression :** **(March 2008; Oct. 2005)**

X = (p + q)/((r + s) + t)

Ans. : The tree structure for the expression

X = (p + q)/((r + s) + t) is as :

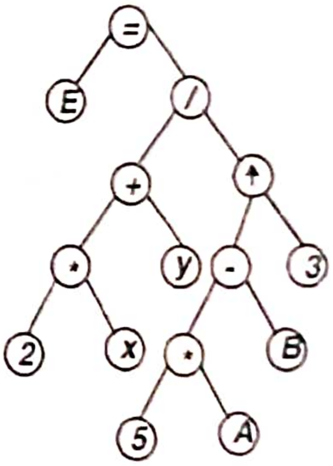


Q. 50 **Draw the tree diagram which corresponds to the following algebraic expression :**

E = (2X + Y)/(5A – B)^3 **(March 2006)**

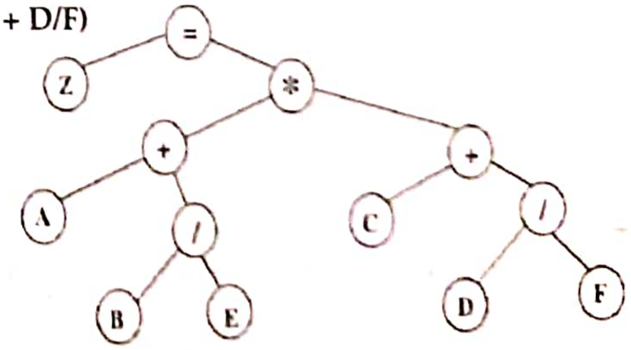
Ans. : Tree diagram which corresponds to the following algebraic expression.

E = (2X + Y)/(5A – B)^3 is as :

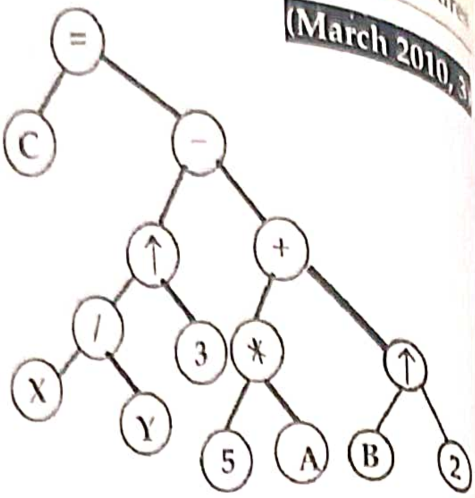


Q. 51 **What is Binary Tree ? Show a tree structure for the expression : Z = (A + B / E) * (C + D/F)** **(Oct. 2007)**

Ans. : Z = (A + B / E) * (C + D / F)



Q. 52 Explain the terms Siblings and Leaf in case of a tree structure Draw tree diagram for the expression $C = (X/Y)^3 - (5A + B^2)$ (March 2010, 3



Ans. :

sibling & Leaf :

(Please Refer Ch2/Q-37/p-2-29)

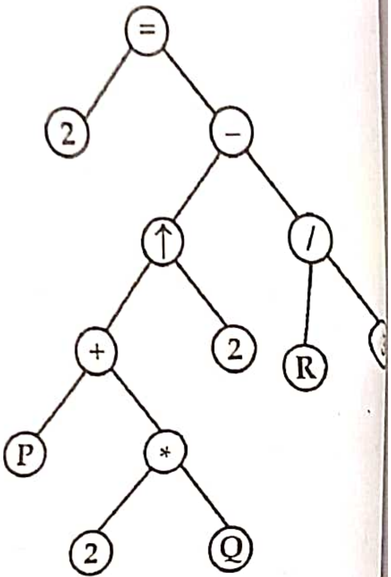
Tree diagram for the expression is as

Q. 53 What is a Tree Structure ? Draw the diagram for the given expression :

$A = (P + 2 Q)^2 - (R/3)$

(Please refer Ch.2, Q.37 (Page No. 2 -29)

(Oct. 2010, 3)



Q. 54 Draw Binary Tree structure for the following expression :

$(2A + B) (5F - D)^3$ (March 2013,

Ans. :

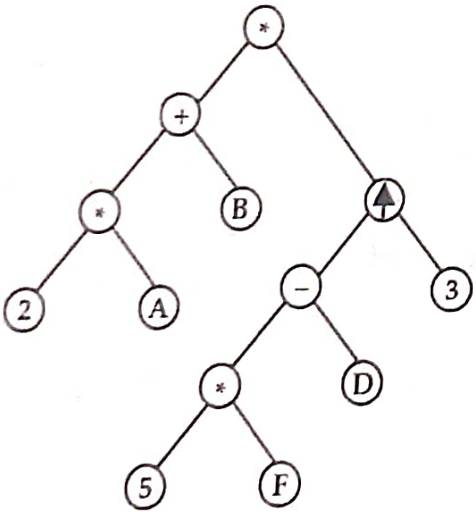


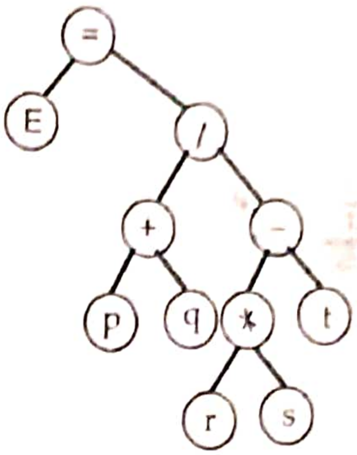
Fig. Q. 54

Q. 55 What is Binary Tree ? Draw the tree structure for the following expression

$E = (p + q) / [(r * s) - t]$ (Oct. 2012,

Ans. :

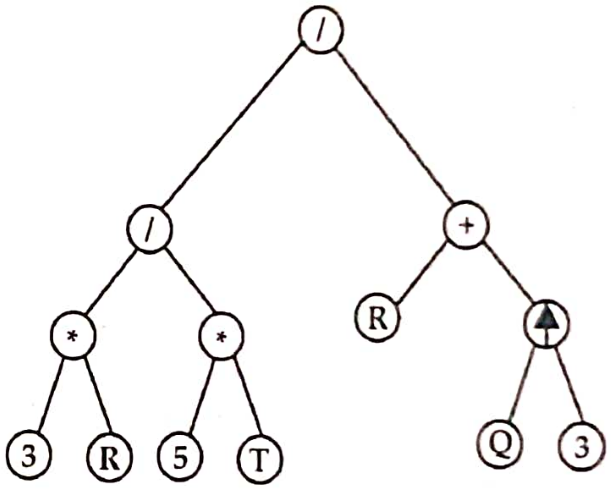
E = (p + q) / [(r * s) – t]



Q. 56 **What is a Binary Tree? Draw tree diagram for the expression**

B = (3R/5T) – (R + Q³) **(Oct. 2013, 4)**

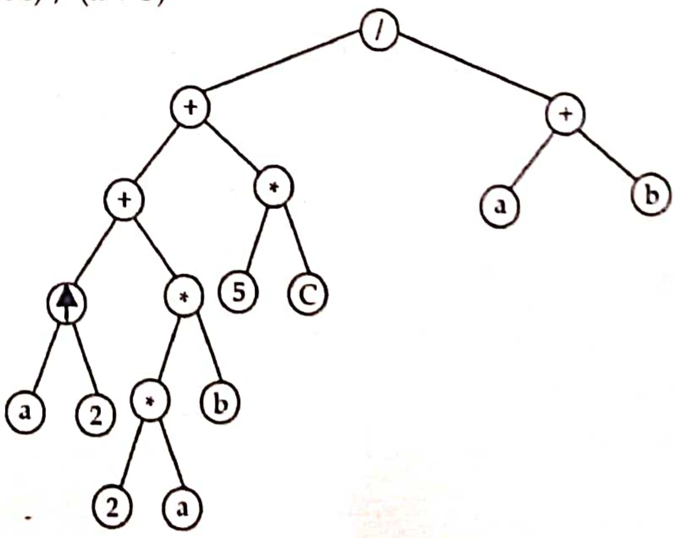
Ans. : Please refer Chapter 2 Q. 39, Pg. No. 2-30.



Q. 57 **Define Binary Tree. Draw Binary tree diagram for the following expression :**
(a² + 2ab + 5c) / (a + b) **(March 2014, 3)**

Ans. : Please refer Chapter 2 Q. 39, Pg. No. 2-31 for Binary tree.

(a² + 2ab + 5c) / (a + b)

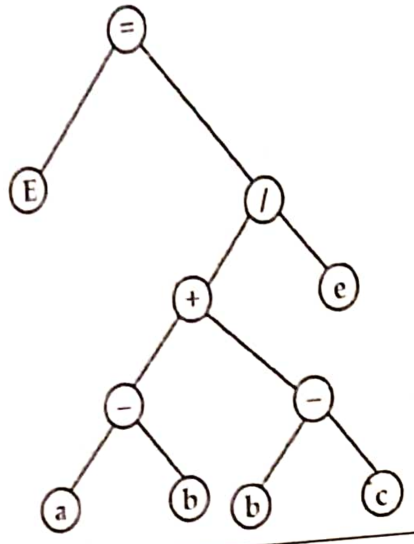


Q. 58 Draw the binary tree for following expression :

E = ((a – b) + (b – c)) / e

(Oct. 2014, 3

Ans. :



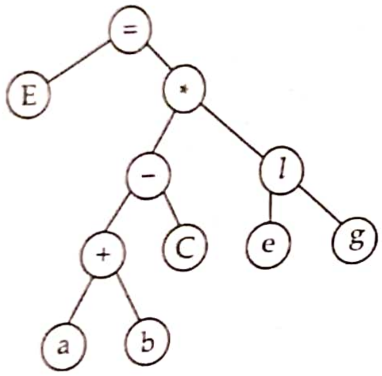
Q. 59 Draw the binary tree for following expression :

E = ((a + b) – c) * (e/g)

(Oct. 2015,

Ans. :

E = ((a + b) – c) * (e/g)

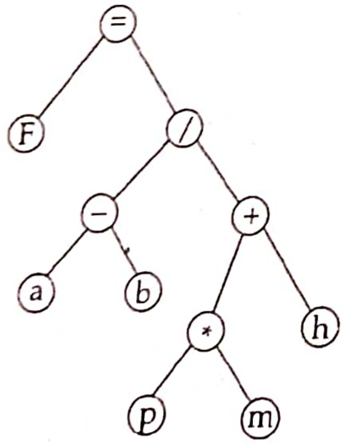


Q. 60 Draw a binary tree structure for the following expression :

F = (a – b) / ((P * m) + h)

(July 201

Ans. :

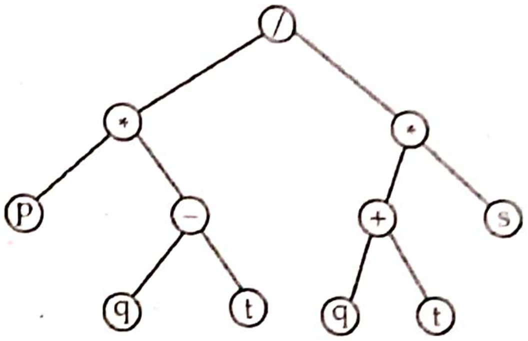


Q. 61 Draw a binary tree structure for the following expression :

(p * (q – t)) / ((q + r) * s)

(Oct. 2006

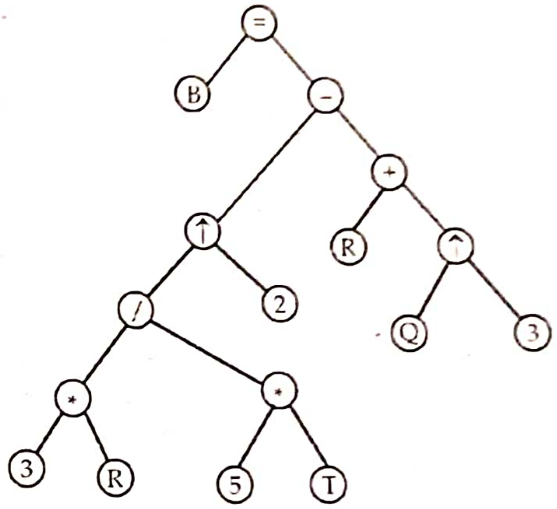
Ans. :



Q. 62 What is Binary Tree ? Draw the Tree diagram for the expression.

$B = (3R/5T)^2 - (R + Q^3)$ (Ch. 2/Q. 40/Pg. No. 2-31) **(March 2019)**

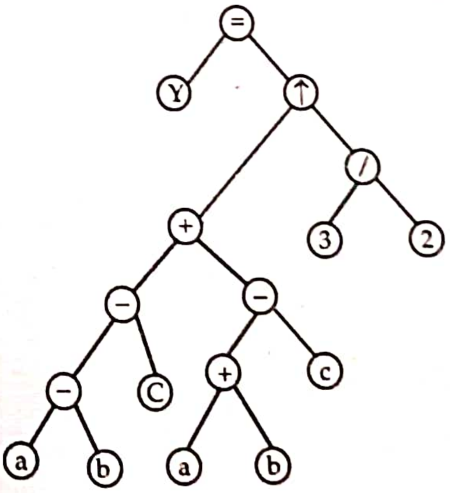
Ans. :



Q. 63 Define Binary tree. Draw a Tree diagram for following expression. **(March 2020)**

$Y= [(a - b - c) + (a + b - c)]^{3/2}$ (Ch. 2/Q. 40 and Q. 63/Pg. No. 2-31 and 2-41)

Ans. :



Q. 64 Select the correct alternative and rewrite the following statements

(March 2005)

1. Finding location of given element is called as ——.
   (i) Traversing (ii) Insertion
   (iii) Searching (iv) None of the above

Ans. : (iii) Searching

(March 2018)

2. Data items that are divided into subitems are called as —
   (i) Group items (ii) Elementary items
   (iii) Nodes (iv) Arrays

Ans. : (i) Group items

3. In LINKED LIST, Link field contains ——.
   (i) Value of next node (ii) Address of next node
   (iii) Value of previous node (iv) None of these

Ans. : (ii) Address of next node

(March 2017

4. A record is a collection of ——.
   (i) Files (ii) Arrays
   (iii) Fields (iv) Maps

Ans. : (iii) Fields

5. The time required to execute bubble sort algorithm having 'n' input items is directly proportional to ——.

(March 2013

   (i) n^2 (ii) n
   (iii) $\text{Log}_2 n$ (iv) $\text{Log}_e n^2$

Ans. : (i) n^2

6. Maximum number of nodes of symmetric binary tree with depth n are ——.

(Oct. 2005, 09, March 2012)

   (i) 2^n (ii) $\text{Log}_2 n$ (iii) n^2 (iv) $2^n - 1$

Ans. : (iv) $2^n - 1$

7. Maximum number of nodes of symmetric binary tree with depth 5 are ——.

(March 2002)

   (i) 5 (ii) 25 (iii) 31 (iv) 32

Ans. : (iii) 31

8. Accessing each element in an array only once is called ——. **(Oct. 2002, March 03, 11)**
   (i) Searching (ii) Inserting
   (iii) Deleting (iv) Traversing

Ans. : (iv) Traversing

9. The elements of record are ——. **(Oct. 2003, March 2009)**
   (i) Homogeneous (ii) Similar
   (iii) Non-homogeneous (iv) Identical

Ans. : (iii) Non-homogeneous

10. The most efficient search algorithm is ——. **(March 2004)**
    (i) Binary search (ii) Reverse search
    (iii) Linear search (iv) Pointer search

Ans. : (i) Binary search

11. The number of comparisons required for bubble sorting of an array of n elements is ————. **(March 2004)**
    (i) $n(n-1)/2$ (ii) $n/2$
    (iii) $\log_2 n$ (iv) $\log_{10} n$

Ans. : (i) $n(n-1)/2$

12. Finding the location of record with a given key value is known as ——. **(March 05; Oct. 11)**
    (i) Traversing (ii) Searching
    (iii) Sorting (iv) Inserting

Ans. : (ii) Searching

13. Maximum number of nodes in a symmetric binary tree with depth four are —— . **(March 2006)**
    (i) 4 (ii) 15
    (iii) 16 (iv) 5

Ans. : (ii) 15

14. In —— data structure, an element may be inserted or deleted only at one end called Top.
    (i) Queue (ii) Array
    (iii) Stack (iv) Tree

Ans. : (iii) Stack

15. Maximum number of nodes of symmetric binary tree with depth of 6 is ——.
    (i) 64 (ii) 6
    (iii) 63 (iv) 36

Ans. : (iii) 63

16. _____ is the only non-linear data structure from the following list.
    (i) Array (ii) Stack
    (iii) Tree (iv) Linked List

Ans. : (iii) Tree

17. _____ is the operation of rearranging the elements of an array either in increasing and decreasing order. **(March 2007)**
    (i) Sorting (ii) Searching (iii) DMS (iv) DBMS

Ans. : **(i) Sorting**

18. The complete binary tree (Tn) has n = 15 nodes then its depth (dn) is _____. **(Oct. 2007)**
    i) 2 ii) 3 iii) 4 iv) 5

Ans. : **(iii) 4**

(March 2008)

19. Maximum number of nodes of symmetric binary tree with depth of 7 is ______

i) 125 ii) 127 iii) 128 iv) 124

Ans. : (ii) 127

(Oct. 2008)

20. Elements of Array are always ______.

(i) Homogenous (ii) Hetrogenous

(iii) Non-homogenous (iv) None of these

Ans. : (i) Homogenous

(March 200

21. Record contains ______ Data.

i) Homogeneous ii) Non-homogeneous iii) Same iv) None of these

Ans. : (i) Homogeneous

(March 201

22. Sorted List is essential requirement for _____ process of an array.

(i) Linear Search (ii) Binary Search (iii) Traversing (iv) Insertion

Ans. : (ii) Binary Search

(Oct. 20

23. Maximum number of nodes of symmetric binary tree with depth 6 are ______

i) 31 ii) 127 iii) 63 iv) 64

Ans. : (iii) 63

(Oct. 20

24. Tree is ______ Data Structure.

(i) Linear (ii) Non-linear

(iii) Homogeneous (iv) Non-homogeneous

Ans. : (iv) Non-linear

(Oct. 201

25. The elements of the binary tree are ______.

(i) Homogenous (ii) Non-homogeneous

(iii) Similar (iv) Identical

Ans. : (ii) Non-homogeneous

(March 201

26. *Complete Binary Tree (T_n) has n= 31 nodes, then its depth is ______:*

(i) 2 (ii) 3 (iii) 4 (iv) 5

Ans. : (iv) 5

(March 2014

27. *Most efficient search algorithm is______.*

(i) Binary (ii) Reverse (iii) Linear (iv) Pointer

Ans. : (i) Binary

(Oct. 2014

28. *Finding location of given element in array is called ______.*

(i) Sorting (ii) Searching

(iii) Traversing (iv) Merging

Ans. : (ii) Searching

29. __________data structure does not require contiguous memory allocation. **(March 2015)**

(i) Array (ii) String (iii) Pointer array (iv) Linked list

Ans. : **(iv) Linked list**

30. Tree is a _______ collection of Nodes. **(Oct. 2015)**

(i) Hierarchical (ii) Linear (iii) Relational (iv) Graphical

Ans. : **(i) Hierarchical**

31. If a complete binary tree (Tn) has n = 1000 nodes, then its depth (Dn) is ___________. **(July 2016)**

(i) 10 (ii) 20 (iii) 50 (iv) 100

Ans. : **(i) 10**

32. __________ is the only non-linear data structure from the following list. **(July 2017)**

(i) Array (ii) Stack (iii) Tree (iv) Linked List

Ans. : **(iii) Tree**

33. If lower bound = 0 and upper bound = 15, then midterm is _____ in binary search method. **(July 2018)**

(i) 6 (ii) 7 (iii) 8 (iv) 9

Ans. : **(iii) 8**

34. __________ is very useful in situation when data is to be stored and retrieved in reverse order. **(March 2019)**

(i) Stack (ii) Queue (iii) Linked List (iv) Tree

Ans. : **(i) Stack**

35. Record contains _________ data. **(July 2019)**

(i) Homogenous (ii) Non-homogenous
(iii) Same (iv) None of these

Ans. : (ii) Non-homogeneous

36. ______ is collection of fields. **(March 2020)**

(i) File (ii) Record
(iii) Array (iv) Queue

Ans. : (ii) Record

❑❑❑

Chapter 3 C++

Scope of the Syllabus

Probable marks : 41

- Review of C++
- Arrays, pointers, references, strings
- Principle of object oriented programming
- Classes and objects
- Constructors and destructors
- Operator overloading and type conversions
- Inheritance
- Virtual functions and polymorphism
- Working with files

REVIEW OF C++

Q. 1 What is C++ ? What are the advantages of C++ ? **(March 2003; 13**

Ans. :

C++ is an object oriented programming language. Initially C++ was named as "C with classes". C++ was developed by Bjarne Stroustrup at AT & T Bell Laboratories, USA, in the early eighties.

The advantages of C++ over C are :

(i) C++ is an incremented version of C. It is a superset of C. Almost all C programs can also run in C++ compiler.

(ii) The important facilities added in C++ are classes, function overloading, operator overloading.

(iii) C++ allow user to create abstract data types, to inherit properties from existing data types.

(iv) C++ supports polymorphism.

(v) Any real life application systems such as editor, compiler, databases, communication systems can be built by C++.

(vi) Object oriented libraries can be built by C++.

(vii) C++ programs can be easily implemented, maintained and expanded.

Q. 2 Differentiate between traditional procedural programming approach and object oriented programming approach.

(Oct. 2002, 2005;-March 2011, March 2018)

Ans. :

The differences between traditional procedural programming approach and object oriented programming approach are as follows :

	Traditional Procedural Programming Approach		Object Oriented Programming Approach
1	In this approach, the problem is viewed as a sequence of things to be done.	1	In this approach, the problem is decomposed into a number of entities called objects and then builds data and function around these entities.
2	Emphasis is on doing things.	2	Emphasis is on the data rather than procedure.
3	Large programs are divided into smaller programs known as functions.	3	Programs are divided into entities known as objects.
4	Data move openly around the system from	4	Data is hidden and cannot be accessed by external functions.
5	Employs top-down approach in program design.	5	Follows bottom-up approach in program design.

Q. 3 What do you mean by Object Based Programming Language and Object Oriented Programming Language ? State the relationship between these languages.

(Oct. 2008, March 2010, 3)

Ans :

Object Based Programming Language :

1) Language that supports programming with objects are said to be object based programming languages.
2) It is a style of programming that primarily supports encapsulation & object or identity.
3) Major features are :
   a) Data encapsulation
   b) Data hiding & access mechanism
   c) Automatic initalization & clean-up objects
   d) operator overloading.
4) They do not support inheritance .& dynamic binding.
5) For eg-Ada.

Object - Oriented Programming Language :

1) This language incorporates all the object based features along with inheritance and dynamic binding.
2) For eg. — C++, Smalltalk.

   The relation between them characterized by following statement :

   Object oriented programming =. object—based features + inheritance + dynamic binding.

...ented programming.

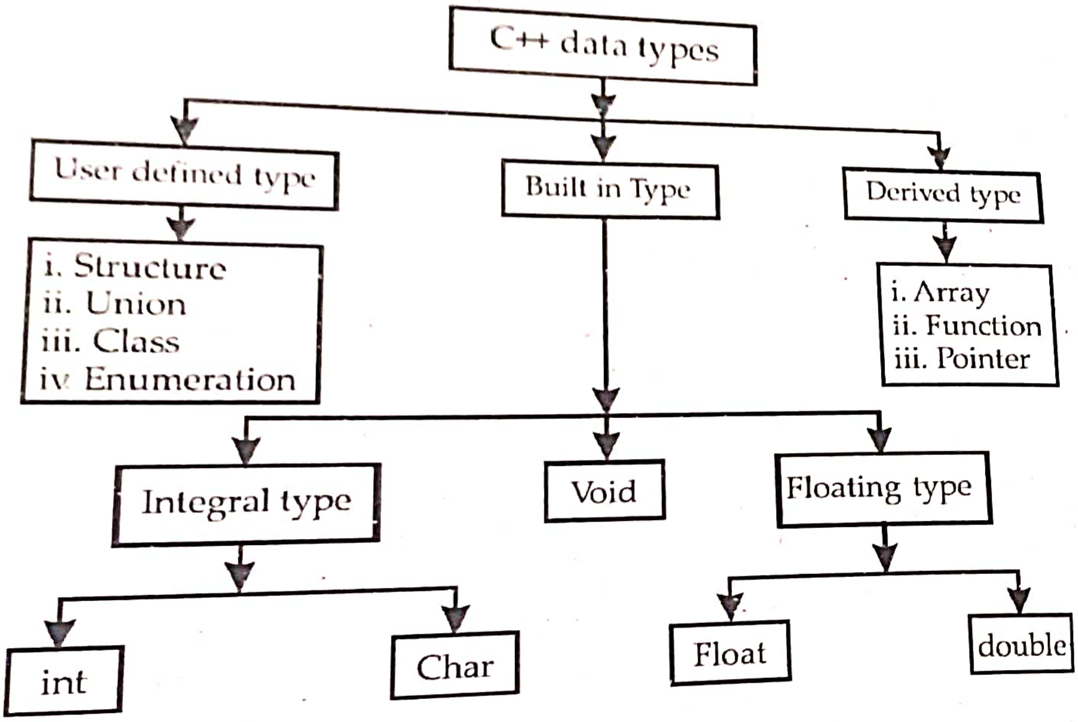
Advantage of Object Oriented Programming: (March 201

1. Through, inheritance, eliminate redundant code and extend the use of existing class
2. The principle of data hiding helps the programmer to build secure program.
3. It is possible, to have multiple instances of ,an object to co-exist without any interfer
4. It is easy to partition the work in a; project based on object.
5. OOP system can be easily upgraded from small to large system.
6. Software complexity can, be easily managed.
7. It is possible to map objects in the problem domain to objects in the program.
8. Good message passing technique for communication between objects.

Q.5 What are the different data types in C++ ? (Oc

Ans. :

1) Data types in C++ are shown in figure below :



2) C++ allows user to create new abstract data types, which can behave like any b data type. These are called user-defined data types. These include structure, union and enumeration.

3) C++ provides three built-in data types which are integral, void and floating.

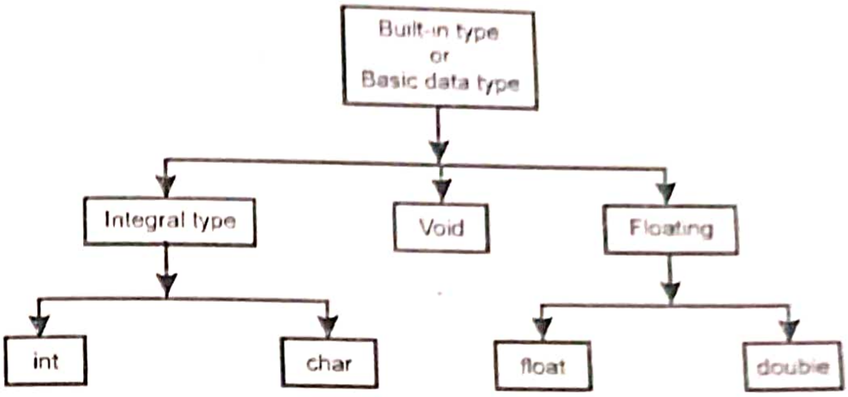
) Integral includes integer and character (string) while floating type includes floa double.

) In addition to these data types, C++ provides user with arrays, functions and poi which are referred as derived data types.

Q. 6 **Enlist the basic data types used in C++ with size of data in terms of bytes for each.** **(March 2002, 2006, October 2006)**

OR **Enlist different built in data types in C++ with their sizes.** **(Oct. 2009)**

Ans. : There are three main basic built-in data types used in C++ viz. integral type, void and floating type.



i) Integral data type :

It includes integer (int) and character (char).

An int variable requires 2 bytes to store, while a character variable requires 1 byte.

Integer variables are also of two types : (a) short int and (b) long int. Long integer requires 4 bytes, while short integer requires 2 bytes.

ii) Void data type :

Void data type is used :

(a) to specify the return type of a function when it is not returning any value.

(b) to indicate an empty argument list to a function.

(c) to declare generic pointers.

iii) Floating type : **(July 2019)**

Floating type variables are of two types; float and double. A float variable requires 4 bytes, while double requires 8 bytes to store in memory.

There is another kind of double namely long double, which requires 10 bytes to store in memory. The following table shows all basic data types, size and range :

Sr. No.	Type	Bytes	Range
1	char (Signed char)	1	– 128 to 127
2	unsigned char	1	0 to 255
3	int (short int or signed int)	2	– 32768 to 32767
4	unsigned int	2	0 to 65535
5	float	4	3.4×10^{-38} to 3.4×10^{-38}
6	double	8	1.7×10^{-308} to 1.7×10^{308}
7	long double	10	3.4×10^{-4932} to 3.4×10^{-4932}

Q. 7 Explain insertion and extraction operators in C++. **(Mar. 2012**

Ans. :

(i) Insertion operator :

The operator "<<" is called as insertion operator. It is also called as "put to" operator. I inserts the contents of the variables on its right to the object on its left.

It is generally used in output statement in C++.

e.g. (i) Cout << a ;

(ii) Cout << "program";

In first example, the value of variable 'a' is printed on screen, while in second exampl the word "program" is printed on screen.

(ii) Extraction operator :

The operator ">>" is called as extraction operator. It is also called as 'get from' operato It extracts or takes the value from keyboard and assigns it to a variable on its right. It used in input statement in C++.

e.g. cin >> a;

This instruction will extract a value from keyboard and assign it to the variable a. C+ allows us to redefine insertion and extraction operators by overloading them.

Q. 8 Write a short note on scope resolution operator. **(Oct. 2014; Mar.1**

Ans. :

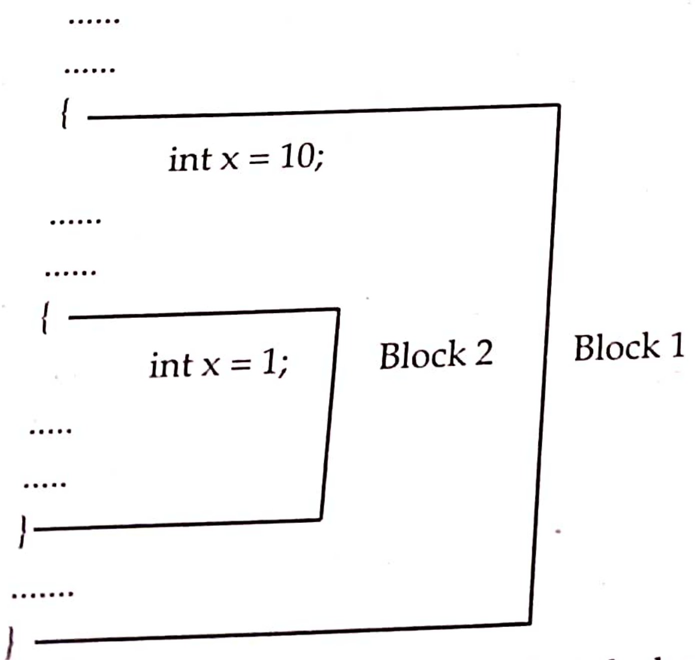
1) The operator :: is called as scope resolution operator.

2) C++ is a block structured language i.e. a C++ program may contain one block with another block.

3) When a variable is declared in program, scope extends from the point of declaration t the end of the block in which it is defined.

4) The same variable name can be used to have different meaning in different blocks.

5) Consider the following segment of program.



```
......
......
{
        int x = 10;
......
......
{
        int x = 1;
.....
.....
}
.......
}
```

Here Block 2 is contained in Block 1. Note that declaration of a variable in an inner bloc hides the declaration of the same variable in an outer block.

6) Scope resolution operator is used to uncover a hidden variable.

It takes the form

```
:: variable name
```

e.g.

```
......
......
{
        int x = 10;
        .......
        {
        int x = 1;
        cout << "Local x is" << x;
        cout << "\n Global x is" << :: x;
........
}
.......
}
```

The output will be as follows :

```
Local x is 1
Global x is 10
```

Q. 9 Explain the use of scope resolution operator and memory management operators in C++ with examples. **(March 2004,16, 17)**

Ans. : Scope resolution operator :

1) In C++, scope resolution operator (: :) is used to access a global variable from a function in which a local variable is defined with the same name as a global variable.

2) For example :

In following program, the function main ( ) access the global variable num and also the local variable with the same name.

```
int num = 20
void main ( )
{
     int num = 10; // local variable
     cout <<"Local =" << num;
     cout <<"Global =" << : : num;
}
```

The output is as :

```
Local = 10
Global = 20
```

Memory management operator :

(1) C++ provides following two memory management operator :

(i) new (ii) delete

(2) The new operator obtains memory block from operating system and returns a pointer to its starting point. The new operator returns NULL, if memory allocation is unsuccessful. The general format of new operator is :

DataType * new DataType [size in integer];

(3) The delete operator is used to return the memory allocated by the new operator back to the memory pool. Thus released memory will be reused by other parts of the program. The general format of delete operator is :

delete pointervariable;

(4) For example :

```
void main ( )
{       char * str = "COMPUTER";
        int len = strlen (str);
        char * ptr;
        ptr = new char [len + 1];
        strcpy (ptr, str);
        cout << "ptr =" << ptr;
        delete ptr;
}
```

In above example, the new operator returns a pointer that point to a memory sectio large enough to hold the string str plus an extra byte for null character. Then after use memory delete operator released memory.

Q. 10 What are the different selection (conditional) statements in C++? Give syntax f each.

Ans.

The program has to be able to evaluate conditions and select alternative path i program.

In C++, there are two ways in which selection may be made :

1) The if statement   2) The switch statement

1) *The if statement :*

The if statement has two forms :

i) Simple if statement   ii) if _____ else statement

i) Simple if statement :

Syntax :

```
    if (condition)
{
    action 1;
}
    action 2;
```

1) Depending on the condition value, program execution proceeds in one direction or another.

2) If the condition is true, then action 1 will be done.

ii) **if .... else statement :**

Syntax :

```
if (condition)
{
action 1;
}
else
{
action 2;
}
action 3;
```

If the condition is true, then and then only action 1 will be done, otherwise action 2 will e done.

) **The switch statement :**

1) This is a multiple branching statement.

2) Depending on certain condition, it executes only one module out of several. If no condition is satisfied, then default module will be executed.

3) The break statement is used to terminate switch statement.

4) Expression must have int or char value.

Syntax :

```
switch (expression)
{
case 1 :
    {
        action 1;
        break;
    }
case 2 :
    {
        action 2;
        break;
    }
:
:
default :
    {
    action x;
    }
}
```

Q. 11 What are the different looping structures in C++ ? Give syntax for each.

Ans. :

Following are the different looping structures in C++ :

1) For loop 2) While loop 3) Do-while loop

1) The for loop :

The for loop is an entry-controlled loop. It is used when action is to be repea predetermined number of times.

Syntax :

```
for (initial expression; test expression; increment / decrement expression)
{ .......
action;
} .......
```

where :

(a) Initial expression is executed only once, when the loop starts.

(b) Test-expression evaluated each time through the loop, before the body of the is executed.

(c) Increment / Decrement expression changes the value of the loop variable end of the loop.

2) The while loop :

The while loop is an entry-controlled loop and it repeats the action until the con becomes false. When condition is false, that time loop is terminated.

Syntax :

```
        while (condition)
{
            action 1;
}
            action 2;
```

3) The do-while loop :

The do-while loop is an exit-control loop used to carry out conditional looping.

Syntax :

```
do
{
            action 1;
}
            while (condition);
            action 2;
```

In do-while, condition is not tested until the body of the loop has been executed once even if the condition is false the loop is executed at least once. If the condition is fa after the first iteration, the loop is terminated.

Q. 12 What is function prototyping ?

Ans. :

1) Function prototyping is one of the major improvements added to C++ functions.
2) The prototype describes the function interface to the compiler by giving details such as the number and the type of arguments and the type of return values.
3) With function prototyping, a template is always used when declaring and defining a function.
4) When a function is called, the compiler uses the template to ensure that proper arguments are passed, and the return value is treated correctly.
5) Any violation in matching the arguments and the return type will be caught by the compiler at the time of compilation itself.
6) Function prototype is a declaration statement in the calling program and is of the following form

```
return-type function-name (argument-list);
```

The argument list contains the types and names of arguments that must be passed to the function.

e.g.

float volume (int x, float y, float z);

Note that each argument variable should be declared independently. The combined declaration like :

float volume (int x, float y, z); is invalid.

7) In function declaration, the names of arguments are the dummy variables and therefore, they are optional i.e. the declaration :-

float volume (int, float, float); is valid.

Q. 13 Write a short note on inline functions.

Ans. :

1) When a function is called, a lot of time is spent in executing a series of instructions, for tasks such as jumping to the function, saving registers, pushing arguments into stack and returning to the calling function.
2) C++ proposes a solution of inline functions to this problem. Inline function makes a program run faster because the overhead of a function call and return is eliminated.
3) However, it makes program to take up more memory, because the statements that define inline function are reproduced at each point where the function is called.
4) "An inline function is a function that is expanded inline when it is invoked". i.e. the compiler replaces function call with the corresponding function code.
5) The inline functions are defined as follows :

```
inline function header
{
function body
}
```

e.g.

```
inline int area (int a, int b)
{
return (a*b);
}
```

6) The functions are generally made inline, when they are small enough to be defined in one or two lines.

7) The keyword inline is not a command, but it is a request to the compiler.

8) Following are some situations in which compiler may ignore inline request :

   i) For functions returning value, if loop, switch or goto statement exists.

   ii) For functions not returning value, if a return statement exists.

   iii) If functions contain static variables.

   iv) If inline functions are recursive.

Q. 14 What are default arguments? Give the advantages of using default arguments.

Ans. :

1) C++ allows to call a function without specifying all its arguments. In such cases, th function assigns a default value to the parameter, which does not have a matchin argument in the function call.

2) Default values are specified when the function is declared.

3) Consider a function area declared as follows,

   float area (int r, float Pi = 3.14);

   The above prototype declares default value 3.14 to the argument Pi. A subsequen function call like -

   A = area(7); //one argument missing

   passes the value 7 to r and lets the function use default value 3.14 for Pi.

   The call A = area (7, 2.5) passes an explicit value 2.5 to Pi.

4) Only trailing arguments can have default values. i.e. add defaults from right to left. A default value cannot provide to an argument in the middle of list.

5) *Advantages of using default arguments :*

   i) These are useful in situations, where some arguments have same values.

   ii) It provides better flexibility to programmers by allowing to use particular arguments that are meaningful to particular solution.

   iii) Use default arguments to add new parameters to the existing functions.

   iv) Default arguments can be used to combine similar functions into a single function.

Q. 15 Explain the concept of function overloading with example.

(March 2008, 15, 17; Oct. 2006

Ans. :

1) The use of same function name to create functions that perform a variety of different tasks is called as function overloading.

2) Overloading refers to the use of same thing for different purposes. Function overloading or function polymorphism, is an example of compile time polymorphism.

3) Using the concept of function overloading, create a family of functions with one function name but with different argument lists.

4) The function would perform different operation, depending on argument list in function call.

5) The correct function to be invoked is determined by checking the number and the type of the arguments and not on the function type.

6) e.g.

```
# include <iostream.h>
int area (int s);              //prototype declaration
int area (int l, int b);       //for overloading area( )
main ( )
{
cout <<area (10);              //function calls
cout <<area (5, 10);
}
int area (int s)               //function definition
{
return (s*s);
}
int area (int l, int b)
{
return (l*b);
}
```

In above example the function area( ) is overloaded. The first function is used to calculate area of square. It has one integer parameter.

The second function is used to calculate area of rectangle. It has two integer parameters.

7) When a function is called, the compiler first matches the prototype having same number and types of arguments and then calls appropriate function for execution. A best match must be unique.

Q. 16 Explain the structure of a general C++ program. **(March 2019)**

Ans. :

1) A typical C++ program contains 4 sections as shown in following figure These sections may be placed in different code files and then compiled independently or jointly.

Include files
Class declaration
Class functions definitions
Main function program

Structure of C++ program

2) It is a common practice to organize a program into three separate files.

3) The class declarations are placed in a header file and the definitions of the member go in other file.

4) This approach enables the programmer to separate the abstract of the interface from the implementation details.

5) Finally the main program that uses the class is placed in third file, which includes the previous two files as well as any other files required.

Q. 17 Write a program in C++ that finds larger number among three numbers.

Ans. :

```
//Program to find largest number
#include <iostream.h>
void main( )
{
int a, b, c, max;
cout<<"Enter three numbers" <<endl;
cin>>a>>b>>c;
if (b>c)
        { max = b;}
else
        {max = c;}
if (a>max)
        {max = a;}
cout<<"The larger number is:-";
cout<<max;
}
```

Q. 18 Write a program in C++ to display a fibonacci series of 15 terms.
(March 2004, 2007, 2009, 2017; Oct. 200

OR Write a program in C++ to display a Fibonacci series of 20 terms (use n < = 18 in th case)

Ans. :

```
#include<iostream.h>
void main( )
{
int f0, f1, f, n;
f0 = 0;
f1 = 1;
clrscr( );
cout<<"Fibbonacci series\n";
cout<<"\n" <<f0<<"\n"<<f1;
for (n=1 ; n<=13; n++)
{
        f = f0 + f1;
        cout<<"\n"<<f;
        f0 = f1;
        f1 = f;
}
}
```

Q. 19 Write a program in C++ to calculate and print factorial of first 10 numbers.

Ans. :

```
//C++ program to calculate and print factorial of first 10 numbers
    #include<iostream.h>
    #include<conio.h>
    void main( )
    {
        int fact, n, i;
        clrscr( );
        cout<<"Number"<<"\t"<<"Factorial";
        for (n=1; n<=10; n++)
        {
            fact = 1;
            for (i = 1; i<=n; i++)
            {
            fact = fact*i;
            }
        cout<<endl<<n<<"\t"<<fact;
        }
    }
```

Q. 20 Write a C++ program to find factorial of a natural number input during program execution. **(March 2004, 08, 17, Oct. 2002,04,12)**

Ans. :

```
//Program to find factorial of a number
#include<iostream.h>
#include<conio.h>
void main ( )
{
    int fact, numberi;
    clrscr ( );
    fact = 1;
    cout << "Enter the number" <<endl;
    cin >> number;
    for (i = 1; i < = number; i++)
    {
    fact = fact * i;
    }
    cout << "The factorial of a inputted number is :" << fact;
}
```

Q. 21 Write a program in C++ to check whether the given integer is palindrome or not. **(Oct.2012)**

Ans. :

```
//C++ program to find whether the given integer is palindrome or not
#include<iostream.h>
#include<conio.h>
```

```
void main( )
{
int n, dn, temp=0, d;
clrscr( );
cout<<“Enter a number”;
cin>>n;
dn=n;
while (dn!=0)
{
        d=dn%10;
        temp=(temp*10)+d;
        dn=dn/10;
}
if (n==temp)
{
cout<<“The number”<<n<<“is palindrome”;
}
else
{
cout<<“The number”<<n<<“is not palindrome”;
}
}
```

Q. 22 What is an armstrong number ? Write a program in C++ to check whether the gi number is armstrong or not.

Ans. : Armstrong number :

“If sum of the cubes of digits of a number is equal to the original number, then t number is said to be an armstrong number”.

e.g. 153 is an armstrong number.

```
//C++ Program to find whether the number is armstrong or not.
#include<iostream.h>
#include<conio.h>
void main( )
{
int n, dn, temp, d;
cout<<“Enter a number”;
cin>>n;
dn = n;
temp=0;
while (dn!=0)
{
        d=dn%10;
```

```
            temp=temp+(d*d*d);
            dn=dn/10;
      }
      if(n==temp)
      {
      cout<<n<<"is armstrong no.";
      }
      else
            {
            cout<<n<<"is not an armstrong number";
            }
      }
```

Q. 23 Write a program in C++ to print the numbers in following manner.

```
1
2  2
3  3  3
4  4  4  4
:
:
n terms
```

Ans. :

```
//C++ program to print given pattern
#include<iostream.h>
#include<conio.h>
void main( )
{
int i, j, n;
clrscr( );
cout<<"Enter a number";
cin>>n;
cout<<endl;
for (i=1; i<=n; i++)
{
        for (j=1; j<=i; j++)
        {
                cout<<i<<"\t";
        }
        cout<<endl;
}
}
```

Q. 24 Write a program in C++ to print the numbers in following manner.

```
1   0   1   0   1
1   0   1   0
1   0   1
1   0
1
```

Ans. :

```
//C++ program to print given pattern
#include<iostream.h>
#include<conio.h>
void main( )
{
clrscr( );
int i, j;
        for (i=5; i>=1; i- -)
        {
                for (j=1; j<=i; j++)
                {
                        cout<<j%2<<“\t”;
                }
                cout<<endl;
                }
        }
```

Q. 25 Write a program to perform arithmetic calculations such as addition, subtraction, multiplication or division, depending on choice using switch statement.

Ans. :

```
//C++ program to generate simple calculator
#include<iostream.h>
#include<conio.h>
void main( )
{
float a, b, result;
int ch;
clrscr();
cout<<“Enter two numbers”;
cin>>a>>b;
cout<<“\n1-addition\n 2-subtraction\n 3-multiplication \n 4-division”;
cout<<“Enter Your Choice :”;
cin>>ch;
switch (ch)
{
```

```
case 1:
        result=a+b;
        cout<<“Sum is”<<result;
        break;
case 2:
        result=a-b;
        cout<<“Difference is”<<result;
        break;
case 3:
        result=a*b;
        cout<<“Product is”<<result;
        break;
case 4:
        result=a/b;
        cout<<“Division is”<<result;
        break;
default:
        cout<<“invalid choice”;
        break;
}
}
```

Q. 26 What is a recursive function ? Write a program in C++ to calculate addition of first n numbers using recursive function.

Ans. : Recursive function :

“A function which is called within the body of the same function itself is called as recursive function.”

```
//Program to calculate addition of first n numbers
        #include<iostream.h>
        #include<conio.h>
        int add (int);
        void main( )
        {
        int n, sum;
        cout<<“Enter a number \n”;
        cin>>n;
        sum=add(n);
        cout<<“Addition of first”<<n<<“numbers is”<<sum;
        }
        //function to calculate addition
        int add (int x)
        {
        int S = 0;
        if (x!=0)
        {
        S=x+add(x-1);
        }
        return(S);
        }
```

Q. 27 Write a program in C++ to calculate volume of cube, cylinder and rectangular box depending on choice by using function overloading.

Ans. : //Program using function overloading

```
        #include<conio.h>
        #include<iostream.h>
        int volume (int s);
        float volume (float r, float h);
        int volume (int l, int b, int h);
        void main( )
        {
        int ch;
        do
        {
                clrscr( );
                cout<<"\n1:Volume of cube";
                cout<<"n2:Volume of cylinder";
                cout<<"\n3:Volume of rectangular box";
                cout<<"\n4:Quit";
                cout<<"\n\n Enter Your Choice";
                cin>>ch;
        switch (ch)
        {
        case 1:
                cout<<"Volume of cube is";
                cout<<volume (5);
                break;
        case 2:
                cout<<"Volume of cylinder is";
                cout<<volume (7.0, 2.0);
                break;
        case 3:
                cout<<"Volume of rectangular box is";
                cout<<volume (3, 5, 7);
                break;
```

```
    case 4: break;
    default :
            cout<<"Invalid choice";
            cout<<"Reenter your choice";
    }
    }
            while (ch!=4);
    }
            //function to calculate volume of cube
            int volume (int s)
    {
            return (s*s*s);
    }
            //function to calculate volume of cylinder
            float volume (float r, float h)
    {
            return (3.14*r*r*h);
    }
            //function to calculate volume of rectangular
            int volume (int l, int b, int h)
    {
            return (l*b*h);
    }
```

Arrays, Pointers, References and Strings

Q. 28 What is an array ? Explain how array can be passed onto a function.

Ans. :

1) "An array is a collection of identical data objects, which are stored in consecutive memory locations under common variable name."

2) Arrays may be one dimensional or multidimensional.

3) The general form for declaration of one-dimensional array is

   data-type array-name [expression];

   e.g. int a[10];

   This declaration creates an array of 10 integers.

4) In general, C++ arrays are zero based. i.e. in above examples, the first array element has index 0 and it is referred as a[0]. Similarly, second array element is a[1] and the last i.e. 10th element is a[9].

5) C++ allows to pass the entire array onto a function.

An array name can be used as an argument for the function declaration. No subscrip brackets are required to invoke a function using arrays.

e.g. float rev (float b[], int c);

This declares a function rev, with two parameters, out of which one is an array.

Q. 29 What are pointers ? Give the advantages of using pointers. (March 11, 19, July 16,

Ans. :

1) "A pointer is a variable, which holds the memory address of other variable."

2) * operator is used to declare pointer in C++. It takes the form as :

datatype * variable name;

e.g. int *ptr;

The above declaration will create a variable ptr, which is a pointer variable and will point to a variable, whose data type is integer.

3) The data type of ptr is not integer, but data type of variable which ptr will po integer.

4) Advantages of using pointers are as :

i) It allows to pass variables, arrays, functions, strings, structures, objects as fu arguments.

ii) It allows to return structured variables from functions.

iii) It supports dynamic allocation and deallocation of memory segments.

iv) By using pointers, variables can be swapped, without physically moving then

v) It allows to establish link between data elements or objects.

Q. 30 What are pointers in C++ ? Explain the use of pointer variables for fu definitions using call by value and call by reference OR

(March 2004,07,08,09, Oct. 20

Explain 'Call by value' and 'Call by reference' with one example of each.

Ans. :

1) **Pointers in C++ :**

A pointer is a variable which holds the memory address of another variable.

* operator is used to declare pointer in C++.

For example : int *ptr;

where ptr is a pointer variable and which will point to a variable whose data ty integer.

2) The use of pointers in a function definition may be classified into two groups :

(1) Call by value (2) Call by reference.

3) **Call by value :**

(a) When a portion of the program invokes a function, control will be transferred fr the main function to the calling function and the value of actual arguments copied to the function.

(b) Within function the actual value may be altered or changed.

(c) When the control is transferred back from function to the program, altered values are not transferred back. This type of passing formal argument to a function is called as call by value

(d) For example :

```
main ( )
{ void funct (int X, int Y);
     .........
funct (X, Y); // Call by value
..........
}
void funct (int a, int b)
{ .........
}
```

4) **Call by reference :** **(Oct.2014)**

(a) In call by reference, when a function is called by a program the address of the actual arguments are copied on to the formal arguments. i.e. the formal and actual arguments are referring to same memory location.

(b) Therefore change in value of formal argument affects the value of actual arguments.

(c) The content of a variable that are altered within the function are return to calling portion of a program in the altered form.

(d) For example :

```
main ( )
{
void funct (int * X, int * Y);
.........
.........
funct (&X, &Y); // Call by reference.
..........
}
void funct (int * a, int * b)
{
.........
}
```

Q. 31 Explain how the memory address of a variable can be accessed in C++.
(March 2004, 07,14 ; Oct. 2004,12)

Ans. :

1) Computer uses memory for storing the values of variables and the memory is a sequential collection of storage cell. Each cell has a number called address of the cell.

2) In C++, if declare a variable, then it gets associated with certain location where the value of the variable is stored.

3) Consider the declaration

int p = 30;

then p → Location name (variable)

$\boxed{30}$ → Value at location

7940 → Location number (address)

Computer has selected 7940 memory location to store the value 30.

4) To access the memory address of a particular variable '&' operator is used. The '& operator returns the memory address of its operand.

For example : a = &p;

assigns the memory address of variable p to the a. This address is the location address variable. The operator '&' is "the address of" operator.

The variable 'a' is declared as pointer variable as that contains the address. The poin variable declared in C++ as,

int * a;

where * indicates that a is a pointer variable.

Q. 32 What is call by reference ? What is the advantage of call by reference over cal value ? **(Mar. 2**

Ans. :

A function can be called by two methods :

(i) Call by value

(ii) Call by reference

1) When a function call passes arguments by value (call by value) the called func creates a new set of variables and copies the values of arguments into them.

2) The function does not have access to the actual variables in the calling program and only work on the copies of values.

3) Provision of reference variables in C++ permits to pass parameters to the function reference.

4) When pass arguments by reference (call by reference) the formal arguments in the cal function become alises to the actual arguments in the calling function. This means t when the function is working with its own arguments, it is actually working on original data.

5) The mechanism of call by value is good, if the function does not need to alter the valu of the original variables in the calling program.

6) But, if a situation to change the values of variables in the calling program. e.g. in bubb sort compare two adjacent elements in the list and interchange them if first is great than second. In such situation, the function should be able to interchange the values variables of calling program, which is not possible by call by value. But it can be done the call by reference method is used.

7) e.g. //Program to interchange the values of variable

```
#include <iostream.h>
void swap (int*, int*); //function declaration
void main( )
{
        int a, b;
        cin>>a>>b;
        swap (&a,&b); //call by reference
        cout<<"a="<<a;
        cout<<"b="<<b;
}
void swap (int*a, int*b) //function definition
{
        int t;
        t=*a; //assign the value at address a to t.
        *a=*b; //put the value at b into a.
        *b=t; //put the value at t into b.
}
```

Q. 33 Explain Library Functions : **(Oct. 2015, 4)**

(i) Strcpy ( ) (ii) Strcmp ( )

Ans. :

(i) **Strcpy ( )**

If S_1 and S_2 are string then strepy (S_1, S_2) copies character string S_2 into character string S_1. It means it creates a duplicate of string S_2.

Char * strcpy (Char * S_1, const char * S_2)

For example

```
int main ( )
{
char S1 [] = "ABCD"
char S2 [] = "XYZ"
cout << "Before strcpy (S1, S2) \ n";
cout << "\t S1 = ["<< S1 <<"], length = " << streln (S1) << endl;
cout << "\t S2 = ["<< S2 <<"], length = " << strln (S2) << endl;
strcpy (S1, S2);
cout <<"After strcpy (S1, S2) \n";
cout << "\t S2 = ["<< S2 <<"], length = " << strlen (S2) << endl;
}
```

```
O/P-Before strcpy (S1, S2)
    S1 = [ABCDE], length = 5
    S2 = [XYZ]; length = 3
    After strcpy (S1, S2)
    S1 = [XYZ], length = 3
    S2 = [XYZ], length = 3
```

(ii) Strcmp ( ) :

Int strcmp (char * S_1, char * S_2);

It compares S_1 with S_2 Returns a negative integer, zero or positive integer according to
is less than equal to or grater than S_2.

Example :

Char *S_1 = "ABCDE"

Char * S_2 = "   "

If (strcmp (S_1, S_2) <0)

  Cout << S_1 << "<" <<S_2 << endl;

  else

  cout <S_1<< ">=" << S_2 << endl;

O/P is → ABCDE > =

Q. 34 Write a C++ program to read 'n' numbers input from keyboard and sort them ascending order. **(Oct. 20**

Ans. : //Program to sort numbers in ascending order

```
#include <iostream.h>
voide main ( )
{
     int a [100];
     int n, i, j, temp;
     cout << "How many numbers ?" <<endl;
     cin >> n;
     cout << "Enter the elements :";
     for (i=1; i<=n; i++)
     {
           cin >> a[i];
     }
     for (i=1; i<=(n–1); i++)
     {
           for (j=1; j<=n–i; j++)
           {
                        if (a[j] > a[j+1])
                 {
                        temp = a[j];
                        a[j] = a[j+1];
                        a[j+1] = temp;
                 }
           }
}
cout<<"Ascending order is :\n";
for (i = 1; i< = n; i++)
{
cout <<a[i] <<endl;
}
}
```

Q. 35 Write a C++ program to sort a set of 10 floats in descending order using bubble sort method. **(March 2003)**

Ans. : Hint : In this program, write instruction as,

```
        if (a[j] < a[j+1])
    instead of
        if (a[j] > a[j+1])
    in above program
```

Q. 36 Write a program in C++, that first initialize an array of ten integers. Then sort that array descending order using sort() by call by reference method.

Ans. : //Program to sort array (use call by reference)

```
#include<iostream.h>
#include<conio.h>
void sort (int*, int*);
void main( )
{
int a[10], i, j;
clrscr( );
cout<<"Enter ten numbers \n";
for (i=0; i<=9; i++)
        {
                cin>>a[i];
        }
cout<<"\n After Sort \n";
for (i=1; i<=9; i++)
{
        for (j = 0 ; j< = (10 -i) ; j++)
        {
        sort (&a[j], &a[j+1]); //call by passing address of variable
        }
}
for (i=0; i<=9; i++) //print sorted array
{
        cout<<a[i]<<"\n";
}
}
//function to sort array
```

```
void sort (int *x, int *y) //function definition
{
        int temp;
        if(*x < *y)
        {
        temp=*x; //assign the value at address x to temp
        *x=*y; //put the value at y into x
        *y=temp; //put the value at temp into y.
        }
}
```

Q. 37 Write the following power ( ) function in C++ that returns X raised to the power where n can be any integer. **(March 2002, 2**

Double power (double X, int p);

Use the algorithm that would compute X20 by multiplying 1 by X 20 times.

Ans. : //Power ( ) function in C++

```
#include<iostream.h>
double power (double, int); //function prototype
void main ( )
{
        double X;
        int n;
        cout << “Enter the value of X\n”;
        cin >> X;
        cout << “Enter the value of n\n”;
        cin >> n;
        cout << “The result is :” <<power (X, n);
}
double power (double x, int n)
{
        if (x = = 0)
                return 0;
        if (n = = 0)
                return 1;
        double y = 1;
        for (int i = 0; i < n; i++)
                y = y * x;
        for (int j = 0; j > n; j– –)
                y = y/x;
        return y;
}
```

Q. 38 Write a C++ program to accept a set of 10 numbers and print the numbers using pointers. **(March 2002, Oct. 2005,12,13, July 2017)**

Ans. : //Program to print numbers using pointers.

```
#include<iostream.h>
#include<conio.h>
void main( )
{
int a[10], i, *ptr;
clrscr( );
cout<<"Enter 10 numbers"<<endl;
for (i=0; i<=9; i++)
{
cin>>a[i];
}
ptr=&a[0]; //or use ptr = &a;
cout<<"\n The numbers are\n";
for (i=0; i<=9; i++)
{
cout<<*ptr<<"\n";
ptr++;
}
}
```

Q. 39 **Write a program in C++ to read a line of text and to count number of words in a text.**

Ans. : //Count words in a line of text **(March 2002, 08, 09; Oct.2013, March 2020)**

```
#include<iostream.h>
#include<string.h>
void main( )
{
        char line [80];
        int count = 1, len, i;
        cout<<"\n Enter a line of text \n";
        cin.getline (line, 80);
        len = strlen (line);
        for (i = 0; i < = len; i++)
{
        if (line [i] = = ' ')
        count++;
}
        cout<<"No. of words are";
        cout<<count;
}
```

Principles of object oriented programming

Q. 40 What is object oriented programming ? Enlist the features of object oriented programming.

(March 2009, Oct.2010, March 201

Ans. : Definition of OOP :

1) "Object oriented programming is an approach that provides a way of modularizin programs by creating partitioned memory area for both data and functions that can b used as templates for creating copies of such modules on demand."
2) In object oriented programming, the program is designed around the data bein operated upon rather than upon the operations themselves.
3) It ties data more closely to functions that operate on it. OOP allows to decompose problem into a number of entities called objects and then builds data and functio around these entities.
4) When a program is executed, the objects interact by sending messages to one another.
5) Each object contains data and code to manipulate the data.

Features of OOP :

(Oct. 2003,11,13 ; March 2005

1) Emphasis is on data rather than procedure.
2) Programs are divided into number of objects.
3) Data structures are designed such that they characterize the objects.
4) Functions that operate on the data of an object are tied together in the data structure.
5) Data is hidden and cannot be accessed by external functions.
6) Objects may communicate with each other through functions.
7) New data and functions can be easily added wherever required.

Q. 41 Explain the following concepts related to object oriented programming :

(i) Objects **(ii) Classes**
(iii) Inheritance **(iv) Polymorphism**
(v) Data Encapsulation **(vi) Data Abstraction** **(vii) Data hiding**

Ans. :

) Objects :

(March 2006, 2008 ; Oct. 2004, 2006,20

1) Objects are the basic runtime entities in object oriented system. for eg. they may represent a person, place, a bank account or any item that t program must handle.
2) Programming problem is analyzed in terms of objects and the nature communication between them.
3) Program objects should be chosen such that they match closely with the real-wor objects.

(Mar. 2006,08,14 ; Oct. 2006,07,1

Classes :

1) Class is a way to bind data and its associated functions together.
2) The entire set of data and code of an object can be made a user defined data ty with the help of a class.
3) In fact an object is nothing but a variable, whose data type is class.
4) Once a class has been defined, user can define any number of objects belonging that class.
5) A class is a collection of objects of similar type.

(iii) **Inheritance :** **(Oct. 2007)**

1) The mechanism of deriving a new class from an existing one is called as inheritance.

2) Inheritance is the process by which objects of one class can acquire the properties of objects of another class.

3) In OOP, inheritance stands for reusability. This means that additional features can be added to an existing class without modifying it.

(iv) **Polymorphism :** **(March 2008, Oct. 2004)**

1) Polymorphism is an important OOP concept. Polymorphism means ability to take more than one form.

2) Polymorphism plays an important role in allowing objects having different internal structures to share the same external interface.

3) This means that a general class of operations may be accessed in the same manner even though specific actions associated with each operation may differ.

4) Polymorphism is extensively used in implementing inheritance.

(v) **Data encapsulation** **(March 2015, 3)**

Encapsulation is the packing of data and functions into a single component. Data encapsulation, also known as data hiding, is the mechanism whereby the implementation details of a class are kept hidden from the user. The user can only perform a restricted set of operations on the hidden members of the class by executing special functions commonly called methods. Encapsulation can be used to hide data member and member function.

(vi) **Data Abstraction** **(March 2015; Oct. 2004)**

1) Abstraction refers to the act of representing essential features without including the background details of explanations.

2) Classes are the concept of abstraction and are defined as a list of abstract attributes and functions to operate on these attributes.

3) They encapsulate all the essential properties of the object that are to be created.

vii) **Data hiding :** **(July 2017)**

(i) Data hiding means keeping details private while giving access to an object only through messages.

(ii) It is hiding unnecessary complexity from outside world which prevents accidental modification.

Classes and Objects

Q. 42 What is a class ? Explain general form of class declaration.
(Mar. 2003,08,11, 13; Oct.2005,11,14; July 2019)

Ans. :

) Class is a way to bind data and its associated functions together.

) It allows the data (and functions) to be hidden, if necessary, from external use.

) When defining a class, a new abstract data type that can be created that treated like any other built-in data type.

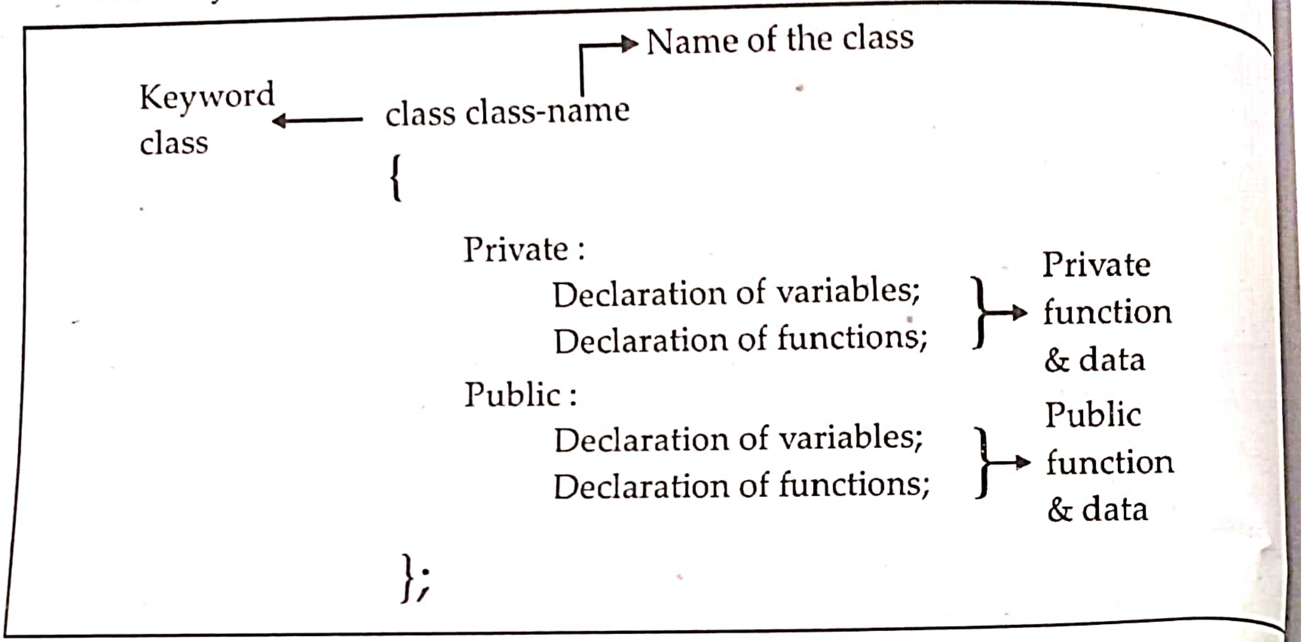
4) Generally, a class specification has two parts :

1. Class declaration. 2. Class functions definitions.

The class declaration describes the type and scope of its members. The class functio definitions describes how class functions are implemented.

5) Class declaration :

Generally a class declaration has following form.



Name of the class

Keyword class

```
class class-name
{
    Private :
        Declaration of variables;
        Declaration of functions;
    Public :
        Declaration of variables;
        Declaration of functions;
};
```

a) The keyword class specifies that what follows is an abstract data type class-name

b) The body of a class is enclosed within braces and terminated by a semicolon.

c) The body of the class contains declaration of class members, which are variab and functions. They are generally grouped under two sections namely **private** a **public**, which are known as visibility labels. These keywords are followed b colon.

d) The members, declared as private can be accessed only from within the class hides data from external use. It is a key feature of OOP.

e) The public members can be accessed from outside the class also.

f) If both the visibility labels are missing, then by default, the members of the class a private. Such a class is completely hidden from outside world and does not ser any purpose.

Example : *A class declaration would look like :*

```
    class item              //specify a class
    {
        int number;         //class data
        float cost;
  public:
        void getdata (int a, float b);
        void putdata (void);
  };
```

Q. 43 Describe how member functions of class can be defined outside the class definition and inside the class definition.

(March 04, 08,12,14 ; Oct. 04,12,15, March 18; July 18, March 2020)

Ans. : Member functions of class can be defined at two places :

(i) Outside the class definition. (ii) Inside the class definition.

Irrespective of the place of definition, the function performs same operation. Hence, code for the function body is identical in both the cases. Only function header is changed.

(i) Outside the class definition :

1) The general form of member function definition outside the class definition is :

```
return-type class-name :: function-name (Argument declaration)
{
function body
}
```

2) The member function incorporates an identity label or membership label (i.e. class-name : :).

3) This label tells compiler the class to which the function belongs and restricts the scope of that function the object of the class 'class-name' specified in header line.

e.g.

```
//class definition
class try1
{
    public:
    void display (void);
};
    //member function definition outside class
    void try1 :: display (void)
{
    cout<<"Programming is fun";
}
```

(ii) Inside the class definition :

1) Another method for defining a member function is to replace the function declaration by the actual function definition.

e.g.

```
class try1
{
    public:
    void display (void)
    {
    cout<<"Programming is fun";
    }
};
```

2) When a function is defined inside a class, it is treated as an **inline function.** Normally, only small functions are defined inside the class definition.

Q. 44 What is an object ? Describe how members of a class be accessed using object of that class. C++

(Oct.2011)

Ans. :

1) An object is a variable whose datatype is a class.

2) User have more than one object for a class. The objects can be declared as :

```
class-name object1-name, object2-name, ...
```

3) The declaration of an object is similar to that of a variable of any basic type. The necessary memory space is allocated to an object at this stage.

Accessing members of class using objects :

1) The private data of a class can be accessed only through the member functions of that class.

2) To use a member function, the dot operator (period) connects the object name and the member function. The dot operator also called as class member access operator.

3) The following format is used for calling a member function :

   objects-name.function-name (actual argument);

4) e.g. Consider a class item defined as follows :

```
            class item
            {
                    private:
                    int number;
                    public:
                    int cost;
                    void getdata (int a, int b) //inline function
                    {
                            number=a;
                            cost=b;
                    }
                    void show (void)
                    {
                    cout<<a<<b;
                    }
        };
                void main( )
        {
                item X;
                X.getdata (20, 30) ;
                ........
        }
```

Here item is a class with private variable number and public members cost, getdata (int, int) and show( ). The X is an object of item.

e.g. X.getdata (20, 30);

This declaration will apply values 20 and 30 to number and cost respectively.

5) Similarly public variables of class can be accessed within main( ) function. But private variables cannot access inside main program, they can be accessed only by public functions of the same class.

Instead of using function as X.getdata (20, 30), if directly apply 20 and 30 to number and cost in main program as :-

```
X.number = 20; //error
X.cost = 20;
```

Then, there will be an error in first statement, because number is private member of class, while there will be no error in second statement, because cost is public member of class.

Q. 45 Write a C++ program to find the Greatest Common Divisor of two numbers. Define a method find to accept the values and calculate GCD of two numbers and print the GCD value. **(March 2002, 2006 ; Oct. 2002, 2005)**

Ans. :

```
//Program to find GCD value
    #include<iostream.h>
    class gcd
    {
            int a, b;
            public :
                    void find ( );
    };
    void gcd :: find (void)
    {
            cout<<"Enter the value of a and b\n";
            cin>>a>>b;
            while (a ! = b)
            {
            if (a > b)
                    a = a – b;
            if (b > a)
                    b = b – a;
            }
                    cout << "The gcd is :" << a;
    }
    void main ( )
    {
            gcd obj1;
            obj1.find ( );
    }
```

Q. 46 What are friendly functions ? Give the characteristics of a friend function. **(Mar.2016 ; 17, 20; Oct.2010, July 2016, 17, 18, 19**

Ans. :

1) C++ allows the common function to be made friendly with more than one classes thereby allowing the function to have access to the private data of classes. Such function need not be a member of any classes.

2) Non-member function cannot have access to the private data of a class. However, there could be a situation where user would like two classes to share a particular function. A this situation friend function is used.

3) To make an outside function "friendly" to a class, simply declare the function as th friend of the class as shown below :-

```
class class-name

{

private:

........

........

public:

........

friend return-type function-name (arguments); //declaration

};
```

4) The keyword "friend' declares the function to be friendly with that class. This functior defined as a normal C++ function. The function definition does not use class-nar keyword friend or scope resolution operator.

5) A friend function has following characteristics :

(i) It is not in the scope of the class to which it has been declared as friend.

(ii) Since it is not in scope of the class, it cannot be called by using object of that class is called like a normal C++ function.

(iii) It can be declared either in public or the private part of a class without affecting meaning.

(iv) Usually, it has the objects as arguments.

(v) It cannot access the member function directly and has to use an object name an dot operator with each member name.

Q. 47 Write any three characteristics of friend function ? **(Oct. 2003, 20**

Ans. : *A friend function has following characteristics :*

(1) It is not in the scope of the class to which it has been declared as friend.

(2) Since it is not in scope of the class, it cannot be classed by using object of that class. It called like a normal C++ function.

(3) It can be declared either in public or the private part of a class without affecting meaning.

Q. 48 Write a program in C++ to demonstrate how a common friend function can be used to exchange the private values of two classes. (Use call by reference method.)

Ans. : //Swapping private data of classes using friend function

```
        #include<iostream.h>
        #include<conio.h>
        class A; //forward declaration
        class B
        {
            private:
            int val2;
            public:
                void getdata (void)
                {
                val2=21;
            }
            void display (void)
                {
                cout<<"Value 2:"<<val2;
                }
            friend void exchange (B&, A&);
            };
        class A
        {
            private:
            int val1;
            public:
            void get (void)
            {
            val1=12;
            }
            void disp (void)
                {
                cout<<"Value 1:"<<val1;
                }
                friend void exchange (B&, A&);
            };
            void exchange (B &x, A &y)
                {
                        int temp;
                        temp = x.val2;
                        x.val2 = y.val1
            y.val1 = temp;
        }
    void main( )
    {
            A abc;
            B pqr;
            abc.get( );
            pqr.getdata( );
            cout<<"Before exchange:-";
            abc.disp( );
            pqr.display( );
            exchange (pqr, abc); //Swapping
            cout<<"\n After exchange:";
            abc.disp( );
            pqr.display( );
    }
```

Note :When a common function is to be made friendly with two classes, forward declaratio necessary, because when we declare friend function in first class, the object of sec function may also be passed as argument to the friend function. But, at that t compiler does not have knowledge about the second class. Therefore an error will ar Hence, forward declaration is necessary.

Q. 49 How members of class are accessed by using pointer object ?

Ans. :

1) Generally in C++, pointer object is used along with normal object and then the addr of normal object is given to pointer object.

2) The pointer object is declared as follow in main() program as :

```
class name object-name, pointer object;
pointer object = address of object name;
```

e.g. *emp e, *ptr;*

ptr=&e;

Here e and ptr are objects of emp, ptr is pointer object, which holds address of e.

3) Now, for access member functions and variables of e, use ptr and '→' (arro operator. Suppose getdata( ) is function of class emp. Then by using pointer object can be accessed as

ptr → getdata( );

Another way of accessing is (*ptr).getdata( );

4) For eg. following program declare a class employee, which contains name, pho number and salary of employee as data. Member functions are used to read data ar display it.

```
#include<iostream.h>
#include<conio.h>
```

```
class emp
{
        private:
        char name[20];
        int ph, sal;
        public:
        void getdata (void);
        void display (void);
};
        void emp::getdata (void)
{
        cout<<"\n Enter employee name-";
        cin>>name;
        cout<<"\n Enter employee salary";
        cin>>sal;
        cout<<"\n Enter phone number-";
        cin>>ph;
}
        void emp::display (void)
{
        cout<<"\n Name:"<<name;
        cout<<"\n Salary:"<<sal;
        cout<<"\n Phone number"<<ph;
}
        void main( )
{
        emp e, *ptr;
        ptr=&e;
        ptr → getdata( );
        ptr → display( );
}
```

Note : This method of using pointer objects in class is same as that of using pointer variables in case of structure.

We can also access array of objects by using single pointer object.

The procedure is

```
emp e[10], *ptr;
ptr=&e[0];
```

This procedure is also same as that in case of structure.

data member in a class.

(March 2005, 11,16,17 ; Oct. 2007,13; July 1

Ans. :

The three special characteristics of a static data member in a class are as follows :

(1) It is initialized to zero when the first object of its class is created. No other initializatio is permitted.

(2) Only one copy of that member is created for the entire class and is shared by all th objects of that class, no matter how many objects are created.

(3) It is visible only within the class, but its life time is the entire program.

Constructor and Destructor

Q. 51 What is a constructor ? Why it is called so ?

(March 2005, 06, 07,11,12,13,14,15; Oct. 2004,05,07,10,12,13,14,1 July 2017, 19, March 18, 19,

Ans. :

1) "A constructor is a special member function of a class. Its task is to initialize the obje of its class."

2) It is special because its name is same as that of the class to which it belongs.

3) The constructor is invoked whenever an object of its associated class is created.

4) It is called constructor because it constructs the values of data members of the class.

5) A constructor can never return any value. Hence, it is written with no return type (ev void is not written).

6) e.g. A constructor is declared and defined as follows

```
        //class with constructor
        class integer
        {
            private:
            int m, n;
            public:
            integer (void); //constructor declared
     };
     integer::integer (void) //constructor defined
     {
          m=0;
          n=0;
     }
```

7) Whenever a class contains a constructor like one above, it will be initialize automatically, whenever an object of that class is created.

i.e. the declaration - integer int1;

This not only creates the object int1 of type integer, but also initializes its data membe m and n to zero.

Q. 52 Give the characteristics of a constructor function. OR What are the syntax rules for writing constructors ?

(Mar. 2003,13, 19; Oct. 2002, 04,12,13, July 2017)

Ans. :

i) The constructor name is always same as the class name.

ii) They do not have return types, not even void and therefore, they cannot return values.

iii) They cannot be static or virtual.

iv) They should be declared in public section.

v) They cannot be inherited though a derived class can call base class constructor.

vi) Like other C++ functions, they can have default arguments.

vii) We cannot refer to their address.

viii) An object with a constructor cannot be used as a member of union.

ix) They make implicit calls to the operators 'new' and 'delete' when memory allocation is required.

x) When a constructor is declared for a class, initialization of class objects become mandatory, since constructor is invoked automatically when the objects are created.

Q. 53 Explain parameterized constructors with default arguments.

Ans. :

1) Generally a constructor initializes the class object to predetermined values. But, in practice, it may be necessary to initialize data elements of objects to different values.

2) C++ permits to achieve this objective by passing arguments to the constructor function when the objects are created.

3) The constructors that can take arguments are called parameterized constructor.

4) For e.g.

```
class fib
{
        private:
        int f0, f1;
        public:
        fib (int x, int y); //Parameterized constructor
};
fib::fib (int x, int y)
{
        f0=x;
        f1=y;
}
```

5) When a constructor has been parameterized, the object declaration statement such as, Fib F; will not initialize the data elements.

6) Pass the initial values to the constructor function when an object is declared. This can be done in two ways :

(i) By calling the constructor explicitly. (ii) By calling the constructor implicitly.

The explicit call can be made as follows.

```
        fib F = fib (0, 1); //Explicitly call
    or  fib F;
        F = fib (0, 1);
```

The implicit call can be made as follows

```
fib F (0, 1); //Implicit call
```

7) User also pass default arguments to the constructor.

```
e.g   class fib
      {
      int f0, f1;
      public :
      fib (int x, int y = 1) //constructor with default arguments
      {
      f0 = x;
      f1 = y;
      }
```

The above declaration sets default value 1 to y.

8) In main ( ), such constructor can be called as

```
          fib F (0, 1);
or        fib F (0); //One argument missing.
```

This call sets f0 to zero and f1 to default value 1.

9) User also call such constructor as -

```
          fib F (0, 2); // No argument missing.
```

This call sets f0 to zero and f1 to two, instead of one.

Q. 54 What is a Constructor ? Explain Copy Constructor with example. **(Oct. 200**

Ans :

Constructor :

A constructor is a special member function of a class. Its task is to initialize the object its class. It is a special because its name is same as that of the class to which it belongs.

Copy constructor

i) Copy constructor are always used when the compiler has to create a temporary objec a class object.

ii) The copy constructors are used in following situations :

  a) The initialization of an object by another object of the same class.

  b) Return of object as by value parameters of a functions.

  c) Stating the object as by value parameters of a functions.

iii) The general format is :
class_name : : class_name (class_name & ptr)

iv) For eg : x : : x (x & ptr)
ptr is a pointer to a class object x.

v) The following program segment shows how to define copy constructor :

```
fib   :: fib ( )        // constructor
{
fo = 0;
f1 = 1;
f = f0 + f1;
}
fib   :: fib (fib & ptr) // copy constructor
{
fo = ptr . fo;
f1 = ptr . f1;
f = ptr . f;
}
```

Q. 55 What is a destructor? Give syntax rules for writing destructor function.
(Mar. 2006,20,09,11,12,14,15; Oct. 2005,07, July 2016, 19, March 2018, 20)

Ans. :

1) A destructor, as the name implies, is used to destroy the objects that have been created by a constructor.

2) The destructor is invoked implicitly by the compiler upon exit from the program to clean up storage that is no longer accessible. In other words, a destructor function gets executed whenever an instance of the class to which it belongs goes out of existence.

3) It is a good practice to declare destructors in a program since it releases memory space for future use.

Syntax rules for writing a destructor function :

(i) A destructor function name is same as that of its class name. But it is preceded by a tilde

e.g. ~fib( ){.... message .....}

(ii) It is declared with no return type since it can never return any value.

(iii) It takes no arguments.

(iv) It should have public access in the class declaration.

Q. 56 What is Constructor and Destructor ? State the difference between them.
(Oct. 2007, March 2011, 3)

Ans : (Ch. 3 / Q. 51 and Q. 55 / Pg. 3-39 and Pg. 3-42)

Difference between constructor & destructor

Constructor		Destructor	
1)	A constructor is a special member of a class. Its task is to initialize'the object of its class.	1)	A Destructor is called to destruction of the object that have been created constructor.
2)	The constructor is invoked whenever an object of its associated class is created.	2)	The destructor is involked important by the complier upon exit from the

Constructor	Destructor
3) A constructor, constructs the values of data members of the class.	3) It is a good practice to declare destructors in a program since it releases memory space fo future

Q. 57 Write a program in C++ to calculate fibonacci series of 'n' numbers usin constructor.

Ans. : //Program to generate fibonacci series

```
#include<iostream.h>
#include<conio.h>
class fibonacci
{
        private:
        long int f0, f1, fib;
        public:
        fibonacci (void);
        void process (void);
        void display (void);
};
fibonacci::fibonacci (void)
{
        f0=0;
        f1=1;
}
void fibonacci::process (void)
{
        fib=f0+f1;
        f0=f1;
        f1=fib;
}
void fibonacci::display (void)
{
        cout<<fib<<"\t";
}
void main( )
{
        int i, n;
        fibonacci F;
        cout<<"\n Enter number of elements"<<endl;
        cin>>n;
        for (i=1; i<=n; i++)
{
        F.process( );
        F.display( );
}
}
```

Q. 58 **Implement a class electricity to calculate electricity bill. The class contains following member functions.**

(i) Getdata ( ) : to get meter number, previous units and current units.

(ii) Process ( ) : to check whether current units are greater than previous unit or not. If not display appropriate message and restart program.

(iii) Calculate ( ) : to calculate electricity bill.

(iv) Display ( ) : to display bill and other details.

Use constructor and destructor.

You may use following rates :

Units	Rates
0-50	Rs 2 per unit
50-200	Rs 3.5 per unit
200-500	Rs 4.5 per unit
500 and above	Rs 5 per unit

Ans. : //Program to calculate electricity bill

```
#include<iostream.h>
#include<conio.h>
class electricity
{
    private:
    int mn, pu, cu, n;
    float bill;
    public:
    electricity( ) // constructor
    {
            bill=0;
    }
    void Getdata (void);
    int Process (void);
    void Calculate (void);
    void Display (void);
    ~ electricity( ) { } // destructor
};
void electricity::Getdata (void) ( )
{
```

```
cout<<"Enter meter No:";
cin>>mn;
cout<<"\n Enter previous units:";
cin>>pu;
cout<<"\n Enter current units:";
cin>>cu;
}
int electricity::Process (void)
{
    if(cu>pu)
    {
            n=cu-pu;
            return(1);
    }
    else
            return(0);
}
void electricity::Calculate (void)
{
   int dn;
   dn=n;
   if (dn<=50)
   {
  bill=bill+(dn*2);
  }
  else
  {
        bill=bill+(50*2);
        if (dn<=200)
        {
       dn=dn-50;
       bill=bill+(dn*3.5);
}
else
{
      bill=bill+(150*3.5);
     if(dn<=500)
     {
    dn=dn-200;
   bill=bill+(dn*4.5);
   }
   else
   {
 bill=bill+(300*4.5);
```

```
                    dn=dn-500;
                    bill=bill+(dn*5);
                    }
                }
            }
        }
            void electricity::Display (void)
            {
                cout<<"\n Meter no:"<<mn;
                cout<<"\n Previous unit"<<pu;
                cout<<"\n Current units"<<cu;
                cout<<"\n No. of units consumed:"<<n;
                cout<<"\n Bill:"<<bill;
            }
            void main( )
            {
                electricity E;
                int a;
            abc:clrscr( );
                E.Getdata( );
                a=E.process( );
                if(a==0)
                {
                cout<<"\n Wrong data Reenter it";
                goto abc;
            }
            else
            {
                E.Calculate( );
                E.Display( );
            }
        }
```

Q. 59 Write a program in C++ to show how multiple constructors can be used in a class.

OR

Write a program in C++ to show overloading of constructors.

Ans. : //Program for overloading constructor

```
            #include<iostream.h>
            #include<conio.h>
            class interest
            {
                    private:
                    float amount, rate, total;
                    public:
```

```
                interest( )        //Default constructor
                {                  //1st constructor
                amount=2000.0;
                rate=10.0;
                }
                interest (float x, float y) //IInd constructor
                {                  //Parameterized constructor
                amount=x;
                rate=y;
                }
        void process (void);
        };
        void interest::process (void)
        {
                total=amount+((rate*amount)/100);
                cout<<"Total="<<total;
        }
        void main( )
        {
                interest I1, I2 (500.0, 5.0);
                cout<<"Default constructor:\n";
                I1.process( );
                cout<<"\n Parameterized constructor:\n";
                I2.process( );
        }
```

Operator Overloading and Type Conversions

Q. 60 What is operator overloading ? Explain with suitable example. OR

(March 2004, 15 Oct.

Explain operator overloading with illustration. Write the advantages of operator overloading.

(Oct.

Ans. :

1) The mechanism of giving some special meaning to an operator is called as oper overloading.

2) In C++, the user defined data types behave in much the same way as the built-in d types.

3) For instance, C++ permits to add two variables of user-defined data types with the s syntax as the basic types. This means that C++ has the ability to provide operators w a special meaning for a data type. This is nothing but operator overloading.

4) Operator overloading provides a flexible option for the creation of new definitions most of the C++ operators.

5) When an operator is overloaded, its original meaning is not lost. For instance, the operator+ has been overloaded to add two vectors, can still be used to add two integers.

6) To define an additional task to an operator, a special function called 'operator function' is used to specify the relation of the operator to the class.

7) Following program shows overloaded ++ operator

e.g.

```
//increment counter variable with ++ operator
#include<iostream.h>
class counter
{
        private:
                int count;
        public:
                counter ( )
                        {count = 0;}
                int get_count ( )
                        {return count;}
                void operator ++ ( )
                        {count ++ ;}
};
void main ( )
{       counter C1, C2;
        cout <<"C1 = "<<C1.get_count ( );
        cout < "C2 = "<<C2.get_count ( );
        C1++;
        C2++;
        ++C2;
        cout <<"C1 = "<<C1.get_count ( );
        cout <<"C2 = "<<C2.get_count ( );
}
```

In the above program, two objects of class counter : C1 and C2 are created. They are initially 0. Then using overloaded ++ operator, increment C1 once and C2 twice and display resulting values.

8) Advantages of operator overloading :

(i) Operator overloading concept extends capability of operators to operate on user-defined data.

(ii) It can also be applied to data conversion.

(iii) Using operator overloading technique, user-defined data types behave in much the same way as the build-in types.

Q. 61 What is operator overloading ? State the three steps involved in operator overloading ? (Mar. 2003,13, 19; Oct. 2008, 10,11,12,13)

Ans. :

1) The mechanism of giving special meanings to an operator is known as operator overloading.

2) Operator overloading provides a flexible option for the creation of new definitions for most of the C++ operators.

3) To define an additional task to an operator, a special function called operator function is used.

4) The process of overloading involves the following steps :

   (a) First create a class that defines the data type that is to be used in the overloading operation.

   (b) Declare the operator function operator op ( ) in the public part of the class. It may be either a member function or a friend function.

   (c) Define the operator function to implement the required operations.

Q. 62 What is an operator function ? Describe the syntax of an operator function. Explain the difference between operator function as member function and friend function. **(Mar. 2002,06,07,11,16, 17, 19; Oct.2010, July 201**

Ans. :

1) To define an additional task to an operator, it specify what it means in relation to the class to which the operator is applied. This is done with the help of a special function called operator function, which describes the task.

2) In short, a function which defines additional task to an operator or which gives a special meaning to an operator is called the operator function.

3) The general form of operator function is,

```
return-type class_name::operator op(argument list)
{
function body//task defined
}
```

Where return type is the type of value returned by the specified operation and op is the operator being overloaded.

The op is preceded by the keyword operator. **Operator op** is the function name.

4) Operator functions must be either member functions (non-static) or friend functions.

5) The basic difference between operator function as a friend function and as a member function is that a friend function will have only one argument for unary operators and two for binary operators, while a member function has no arguments for unary operators and only one for binary operators. This is because the object used to invoke the member function is passed implicitly and therefore is available for the member function. This is not the case with friend function. Arguments may be passed either by value or by reference.

Q. 63 State any eight rules for overloading operators. **(Mar.13, 19; Oct. 07,10,12,15, July 1**

Ans. :

There are certain restrictions and limitations for overloading operators. Some of them are listed below :

1) Only existing operators can be overloaded. New operators cannot be created.

2) The overloaded operator must have atleast one operand that is of user-defined type.

3) The basic meaning of an operator cannot change. i.e. we cannot redefine the plus (+) operator to subtract one value from the another.

4) The overloaded operators follow the syntax rules of original operators.

5) Following are some operators that cannot be overloaded.

Size of	Size of operator
•	Membership operator
•*	Pointer to member operator
: :	Scope resolution operator
? :	Conditional opeator

6) Following certain operators cannot be overloaded using friend functions but member functions can be used to overload them.

=	Assignment operator
( )	Function call operator
[ ]	Subscripting operator
→	Class member access operator

7) Unary operators, overloaded by means of a member function, take no explicit arguments and return no explicit values.

8) Unary operators, overloaded by means of a friend function take one reference argument.

9) Binary operators overloaded through a member function take one explicit argument.

10) Binary operators overloaded through a friend function takes two explicit arguments.

11) When using binary operators overloaded through a member function, the left hand operand must be an object of the relevant class.

12) Binary arithmetic operators such as +, –, * and / must explicitly return a value. They must not attempt to change their own arguments.

Q. 64 Enlist the operators which cannot be overloaded and the operators where friend functions cannot be used.

Ans. :

1) Operators which cannot be overloaded :

(i)	sizeof	- Size of operator.
(ii)	.	- Membership operator
(iii)	.*	- Pointer to member operator.
(iv)	::	- Scope resolution operator.
(v)	?:	- Conditional operator.

2) Operators where friend function cannot be used :

(i)	=	- assignment operator
(ii)	( )	- function call operator
(iii)	[ ]	- subscripting operator
(iv)	→	- class member access operator

Q. 65 Write a short note on type conversions.

Ans. :

1) When constants and variables of different types are mixed in an expression, compil applies automatic type conversions to the operands as per certain rules.

2) Similarly, an assignment operator also causes the automatic type conversions.

3) The type of data to the right of an assignment operator is automatically converted to th type of variable on the left.

4) e.g.

```
int x;
float y;
y=29.123;
x=y;
```

These statements convert y to an integer before its value is applied to x. Thus, t fractional part is truncated.

5) The type conversions are automatic as long as data types involved are built-in types.

6) Consider the following statement that adds two objects and then assigns the result t third object.

V3 = V1 + V2; //V1, V2, V3 are class type objects.

When the objects are of same class type, the operations of addition and assignment carried out smoothly and compiler makes no complaint.

7) But, if one operand is an object and other is built-in type variable or the objects are different classes, then compiler gives error.

8) Since, the user-defined data types are designed by us to suit our requirements, t compiler does not support automatic type conversions for such data types. Therefo design the conversion routines, if such operations are required.

9) Three types of situations may arise in the data conversion between uncompatible types
   1. Conversion from built-in type to class type.
   2. Conversion from class type to built-in type.
   3. Conversion from one class type to another class type.

Q. 66 Explain the three types of data conversion in C++ with a suitable example.
(March 2005, 10, 19; July I

Ans. : Three types of data conversion in C++ are as follows :

(i) Conversion from built-in type to class type.

(ii) Conversion from class type to built-in type.

(iii) Conversion from one class to another class.

(i) Basic to class type :

The constructor can be used for default type conversion from argument's type to th constructor's class type.

```
For e.g.
class time
{
int hr;
int min;
public :
```

```
time (int t) // constructor
{
hr = t / 60; //t in minutes
min = t % 60;
}
};
```

The following conversion statements can be used in a function.

```
time T1; // object T1 is created.
int duration = 90;
T1 = duration; // int to class type
```

After this conversion, the hr member of T1 will contain a value of 1 and min contain 30 means 1 hour and 30 minutes.

(ii) **Class to basic type :**

Overloaded casting operator is used to convert a class type data to basic type. The general form is as :

```
operator typename ( )
{
......... (function statement)
.........
}
```

The conversion function must satisfy following conditions :

(a) It must be a class member.

(b) It must not specify a return value.

(c) It must not have any arguments.

For e.g.

```
Time : : operator int ( )
{
int min1 = hr * 60;
min1 = min1 + min;
return min1;
}
```

The operator int ( ) can be used as follow :

```
Time T1 ; // T1 object
int m = T1; // Class to basic
```

After the conversion 1 hr 30 mins can be converted into 90 minutes.

(iii) **One class to another class type :**

Use one-argument constructor or conversion function depends upon the defining conversion routine in source class or destination class.

For e.g. :

```
obj A        =    obj B
  ↓                 ↓
Destination       Source
```

Constructor is placed in the destination class and conversion function is placed in source class.

Q. 67 Write a program in C++ to overload unary minus operator, so that unary minus operator when applied to an object should change the sign of each of its data items.

Ans. : //Negate all data items of class with - operator

```
#include<iostream.h>
#include<conio.h>
class space
{
        private:
        int x, y, z;
        public:
                space (int a, int b, int c)
                {
                x=a;
                y=b;
                z=c;
                }
                void display (void);
                void operator-( ); //overloaded unary-
                };
                void space::display( )
                {
                cout<<x<<"\t"<<y<<"\t"<<z;
                }
                void space::operator-(void)
                {
                x=-x;
                y=-y;
                z=-z;
                }
                void main( )
                {
                int l, m, n;
                space S;
                cout<<"Enter three numbers \n";
                cin>>l>>m>>n;
S=space (l, m, n);
cout<<"S:";
S.display( );
- S;
cout<<"-S:";
S.display( );
}
```

Q. 68 Write a program in C++ to overload binary + operator for addition of two complex numbers.

Ans. : //C++ program to overload binary + operator for addition of two complex numbers.

```
#include<iostream.h>
#include<conio.h>
class complex
{
        private:
        float x;        //real part
        float y;        //imaginary part
        public:
        complex (float, float);
        complex operator + (complex);
        void display (void);
};
complex::complex (float real, float imag)
{
        x=real;
        y=imag;
}
void complex::display (void)
{
        cout<<x<<"+"<<y<<"i\n";
}
complex complex::operator+(complex c)
{
        complex temp;
        temp.x=x+c.x;
        temp.y=y+c.y;
        return(temp);
}
void main( )
{
        complex C1, C2, C3;
        C1=complex (3.5, 2.5);
        C2=complex (1.1, 1.7);
        C3=C1+C2;               //operator function invoked
cout<<"C1=";
C1.display( );
cout<<"C2=";
C2.display( );
cout<<"C3=";
C3.display( );
}
```

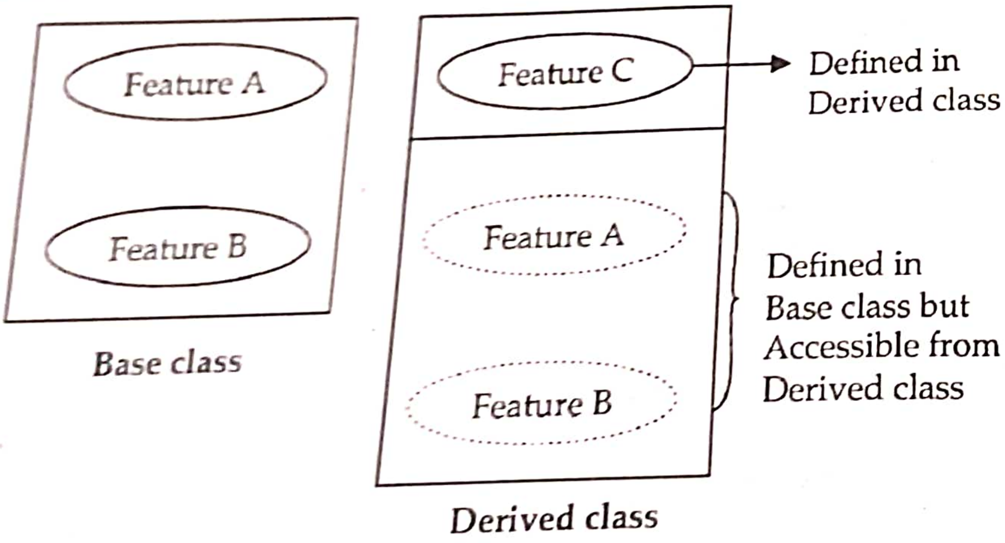
INHERITANCE

Q. 69 What is inheritance ? Explain with suitable example.

(Mar.2015; Oct. 2004,06,08,10,11, July 2016)

Ans. :

1) The mechanism of deriving a new class from an old one is called as inheritance.
2) The old class is referred as base class and new class is referred as derived class.
3) C++ strongly supports the concept of reusability. Once a class has been written and tested, it can be adapted by other programmers to suit their requirements.
4) This is basically done by creating new classes, reusing the properties of the existin ones.
5) Functions and variables of a class that has been tested can be used by object of anoth class. This is known as inheritance.
6) The reuse of a class that has already been tested, debugged and used many times, ca save the efforts of developing and testing the same again.
7) Figure shows inheritance :



Base class

Derived class

8) The syntax of declaration of derived class is :

```
class derived class_name:visibility_mode base_class_name
{
...
...          // Members of derived class
...
};
```

where visibility mode is optional and if present may be either private or public.

```
class base
{
        public :
        void showbase ( )
        {
        cout<<"This is the base";
        }
};
class derived:public base //Declaration of derived class
{
        public:
        void showderived (void)
        {
        showbase ( ); //Base class function used
        cout<<"\n This is derived class";
        }
};
```

Q. 70 Explain different types of inheritances with suitable diagram.

(March 2003,05,09,14,15, 17 ; Oct. 2002,04,06,08,10,11, July 2016, 19)

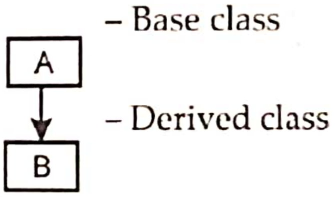
Ans. :

There are five types of inheritances in C++ :

(i) **Single inheritance :**

A derived class with only one base class is called as single inheritance.

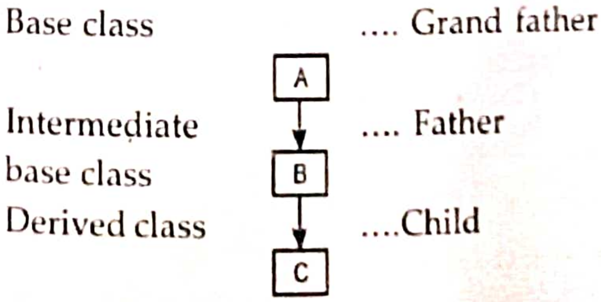
It has the form :



(ii) **Multilevel inheritance :**

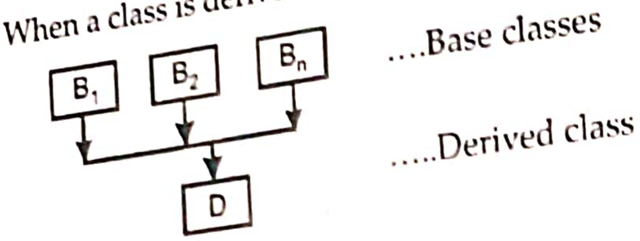
The mechanism of deriving one class from another derived class is multilevel inheritance.

It has the form :



(iii) **Multiple inheritance :**

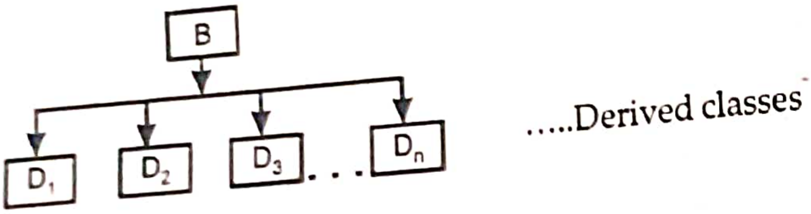
When a class is derived from several base classes, it is called as multiple inheritance.



(iv) **Hierarchical inheritance :**

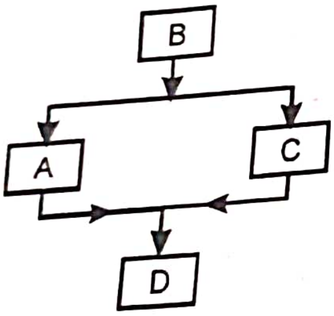
The traits of one class may be inherited by more than one classes. This process is know as hierarchical inheritance.

It has the form :



(v) **Hybrid inheritance :**

The inheritance which involves more than one inheritances is called as hy inheritance. For e.g. : Above figure involves hierarchical, multiple and multi inheritances and the resultant inheritance is called hybrid inheritance.



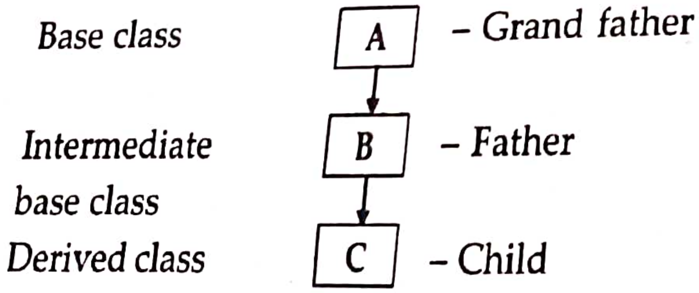
Q. 71 Explain multilevel and multiple inheritance in detail.

Ans. :

1) ***Multilevel inheritance :***

The mechanism of deriving one class from another derived class is called as multilev inheritance.

Following figure shows multilevel inheritance.



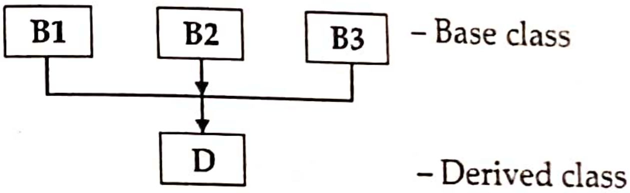
The class A serves as a base class for derived class B which in turn serves as a base class for the derived class C. The class B is known as intermediate base class since it provides link for the inheritance between A and C. The chain ABC is known as inheritance path.

A derived class with multilevel inheritance is declared as follows -

```
class A              //Base class
{---
};
class B:public A     //B derived from A
{---
};
class C:public B     //C derived from B
{---
};
```

2) **Multiple inheritance :**

A class can inherit the attributes of two or more classes (as shown in figure). This is known as multiple inheritance.



Multiple inheritance allows us to combine the features of several existing classes as a starting point for defining new classes.

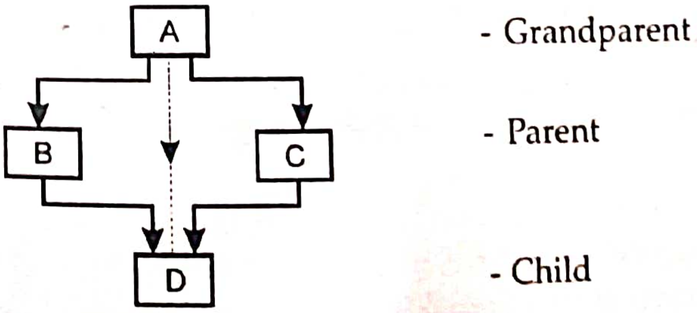
The syntax of a derived class with multiple base classes is as follows :

```
class D:visibility B1, visibility B2, ....
{
(Body of class D)
};
```

where, visibility may be either public or private. The base classes are separated by commas.

Q. 72 What is virtual base class ? Why is it necessary to define virtual base classes in some cases of hybrid inheritance ?

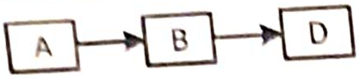
Ans. :



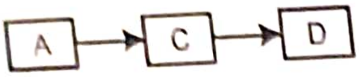
1) Sometimes, when hybrid inheritance is used, there are at least three levels as shown in below figure Here, hierarchical, multiple and multilevel inheritances are used to implement hybrid inheritance.

2) In figure classes B and C are derived from class A and class D is derived from classes B and C. So, class D can inherit members of class A by two paths :

Path Ist : -



Path IInd :-



3) Class D can also inherit members of A directly as shown in figure by dotted line. Th grand parent (i.e. class A) is sometimes referred as indirect base class.

4) This means that class D may contain duplicate sets of members of class A i.e. th members of class A are inherited in class D twice via class B and via class C, Th produces ambiguity.

To avoid this ambiguity, concept of virtual base class is used.

5) Thus, the duplication of inherited members due to multiple paths can be avoided b making the common base class (ancestor class or grand- parent) as virtual base cla while declaring the direct or intermediate base classes as follows :

```
class A //grandparent
{.....
};
class B:virtual public A      //parent 1
{.....
};
class C:public virtual A      //parent 2
{.....
};
class D:public B, public C   //child
{......    //only one copy of A will be
......     //inherited
};
```

6) When a class is made a virtual base class, C++ takes necessary care to see that only one copy of that class is inherited, regardless of how many inheritance paths exist betwee the virtual base class and the derived class.

7) The keywords virtual and public may be used in either order.

Q. 73 What is single inheritance ? Write a program to implement single inheritance.

Ans. :

1) A derived class with only one base class is called single inheritance.

2) For e.g.

Here B is base class and D is derived class. The class B contains one private data member, one public data member and three public member functions. The class D contains one private data member and two public member functions.

```
//Single Inheritance
        #include<iostream.h>
        #include<conio.h>
        class B                    //Base class
        {
        private:
                int a;             //Private member, not inheritable.
        public:
                int b;             //public; ready for inheritance
                void get_ab (void);
                int get_a (void);
                void show_a (void);
        };
        class D:public B //Derived class (public derivation)
        {
                private:
                int c;
                public:
                void mul (void);
                void display (void);
        };
        //....Functions definition .....
        void B::get_ab (void)
        {
                a=5; b=10;
        }
                int B::get_a (void)
        {
                return (a);
        }
                void B::Show_a (void)
        {
                cout<<"a="<<a<<endl;
        }
                void D::mul (void)
        {
                c=b*get_a( );
        }
                void D::display (void)_
        {
                cout<<"a="<<get_a( );
                cout<<"\n b="<<b;
                cout<<"\n c="<<c;
        }
                //....main program ....
        void main ( )
        {
                D d;
                d.get_ab( );
                d.mul( )
                d.display( );
        }
```

The output of above program is :

```
a = 5
b = 10
c = 50
```

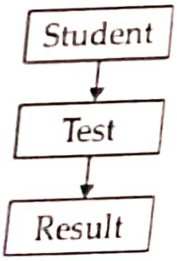
Q. 74 What is multilevel inheritance ? Write a program in C++, to implement multileve inheritance :

Ans. :

The mechanism of deriving one class from another derived class is called as multileve inheritance.

For eg.

Here student is a class, which stores roll number. Class test stores the marks obtained in two subjects and class result contains total marks obtained in test. The class result inherits the details of the marks obtained in the test and roll number through multilevel inheritance.



```
//Multilevel Inheritance
        #include<iostream.h>
        #include<conio.h>
        class student //Base class
        {
        protected:
                int roll_number;
        public:
                void get_number (int);
                void put_number (void);
```

```
};
class test:public student      //Intermediate
{                              //base class
        protected:
        float sub1;
        float sub2;
        public:
        void get_marks (float, float);
        void put_marks (void);
};
class result:public test //Derived class
{
private:
        float total;
public:
        void display (void);
};
        //.....functions definition.....
        void student::get_number (int a)
{
        roll_number=a;
}
        void student::put_number (void)
{
        cout<<"Roll number"<<roll_number<<"\n";
}
        void test::get_marks (float x, float y)
{
        sub1=x;
        sub2=y;
}
        void test::put_marks (void)

{
        cout<<"Marks in sub1="<<sub1<<"\n";
        cout<<"Marks in sub2="<<sub2<<"\n";
}
        void result::display (void)
{
        total=sub1+sub2;
        put_number( );
```

```
            put_marks( );
            cout<<"Total="<<total<<"\n";

    }
            void main( )
    {
            result student1;
            student1.get_number (127);
            student1.get_marks(98, 99.2);
            student1.display( );
    }
```

Q. 75 What is multiple inheritance ? Write a program to implement multiple inheritance

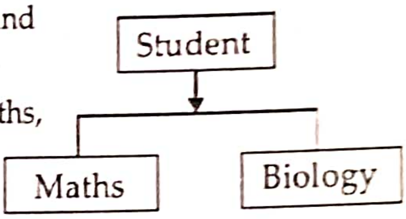
Ans. : "The derivation of one class from several base classes is called as multiple inheritance

```
    //Multiple Inheritance
        #include<iostream.h>
        #include<conio.h>
        class M          //Parent Ist
        {
        protected:
                int m;
        public:
                void get_m(int a)
                {
                m=a;
                }
      };
      class N          //Parent IInd
      {
      protected:
      int n;
      public:
      void get_n(int b)
              {
              n=b;
              }
    };
    class P:public M, public N //child
    {
            public:
            void display (void);
    };
            void P::display (void)
        {
                cout<<"\n m="<<m;
                cout<<"\n n="<<n;
                cout<<"\n m*n="<<m*n;
        }
                void main( )
                {
                P p;
                p.get_m(10);
                p.get_n(20);
                p.display( );
                }
```

Q. 76 What is hierarchical inheritance ? Write a program to implement hierarchical inheritance.

Ans. : "The inheritance, in which two or more classes are derived from same base class is called as hierarchical inheritance."

For e.g.



Here student is a class, which stores the name of student and roll number and marks obtained by student in physics and chemistry. Class maths contain marks obtained in maths, which is derived from class student and calculates PCM grouping. Class Biology, derived from student contains marks obtained in biology and calculates PCB grouping of student.

```
    //Hierarchical Inheritance
        #include<iostream.h>
        #include<conio.h>
        class student
        {
        protected:
                char name[30];
                int phy, che, roll;
        public:
                void getdata (void);
                void display (void);
        };
                class maths:public student
        {
        protected:
                int math;
                float pcm;
        public:
                void get_maths (void);
```

```
          void show_maths (void);
};
          class biology: public student
{
protected:
          int bio;
          float pcb;
public:
          void get_bio (void);
          void show_bio (void);
};
          void student::getdata (void)
{
          cout<<"Enter name and roll number";
          cin>>name; cin>>roll;
          cout<<"\n Enter marks in physics:";
          cin>>phy;
          cout<<"\n Enter marks in chemistry:";
          cin>>che;
}
          void student::display (void)
          {
          cout<<"Name:"<<name;
          cout<<"\n Roll No:"<<roll<<endl;
          }
          void maths::get_maths (void)

          {
          cout<<"\n Enter marks in maths:";
          cin>>math;
          pcm=(phy+che+math)/3;
          }
          void maths::show_maths(void)
          {
          cout<<"\n Marks in physics:"<<phy;
          cout<<"\n Marks in chemistry:"<<che;
          cout<<"\n Marks in maths:"<<math;
          cout<<"\n PCM grouping:"<<pcm<<endl;
          }
          void biology::get_bio (void)
          {
          cout<<"Enter marks in biology:";
          cin>>bio;
          pcb=(phy+che+bio)/3;
          }
          void biology::show_bio (void)
```

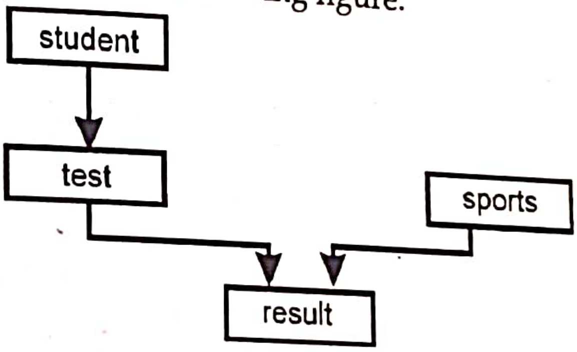
```
        {
        cout<<"\n Marks in physics:"<<phy;
        cout<<"\n Marks in chemistry:"<<che;
        cout<<"\n Marks in biology:"<<bio;
        cout<<"\n PCB grouping:"<<pcb;
    }
        void main( )
        {
        maths M;
        biology B;
        M.getdata( );
        M.get_maths( );
        B.get_bio( ); clrscr( );
        B.display( );
        M.show_maths( );
        B.show_bio( );
        }
```

Q. 77 What is hybrid inheritance ? Write a program to implement hybrid inheritance.

Ans. : "The inheritance, which involves two or more other inheritances is called hybrid inheritance."

For e.g.

Here student is base class of test. Class result is derived from classes test and sports. The class relationships are shown in following figure.

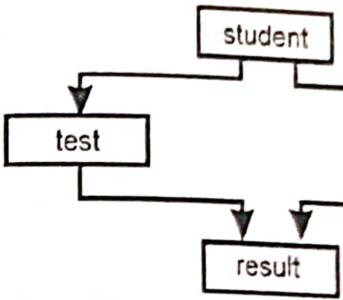


```
//Hybrid Inheritance
        #include<iostream.h>
        class student
        {
        protected:
                int roll_number;
        public:
                void get_number (int a)
                {
                roll_number=a;
                }
                void put_number (void)
                {
                cout<<"\n Roll No:"<<roll_number;
```

```
            }
        };
            class test:public student
            {
            protected:
            float part1, part2;
        public:
            void get_marks (float x, float y)
            {
            part1 = x;
            part2 = y;
            }
            void put_marks (void)
            {
            cout<<“Marks obtained:\n”;
            cout<<“part1=”<<part1;
            cout<<“\n part2=”<<part2;
            }
        };
            class sports
        {
        protected:
            float score;
        public:
            void get_score (float S)
            {
            score = S;
            }
            void put_score (void)
            {
            cout<<“\n Score=”<<score;
            }
    };
            class result:public test, public sports
    {
    protected:
            float total;
    public:
            void display (void);
    };
            void result::display (void)
            {
            total=part1+part2+score;
            put_number( );
            put_marks( );
            put_score( );
```

```
cout<<"\n Total="<<total;
}
void main( )
{
result R;
R.get_number (127);
R.get_marks(83.0, 83.0);
R.get_score(83.0);
R.display( );
}
```

Q. 78 Write a program to implement hybrid inheritance using virtual base class. The class relationships are shown in following Figure.



Class student contains roll number of student. Class test contains marks obtained in two subjects and class sports contains score. Class result should calculate total and display it.

Ans. :

```
//Virtual Base Class
            #include<iostream.h>
            class student
            {
            protected:
                    int roll_number;
            public:
                    void get_number (int a)
                    {
                    roll_number=a;
                    }
                    void put_number (void)
                    {
                    cout<<"\n Roll No:"<<roll_number;
                    }
            };
                    class test:virtual public student
            {
            protected:
                    float part1, part2;
            public:
                    void get_marks (float x, float y)
                    {
```

```
        part1=x;
        part2=y;
        }
        void put_marks (void)
        {
        cout<<"\n Part1="<<part1;
        cout<<"\n Part2="<<part2;
        }
};
        class sports:public virtual student
{
protected:
        float score;
public:
        void get_score (float s)
        {
        score=s;
        }
        void put_score (void)
        {
        cout<<"\n Score:"<<score;
        }
};
        class result:public test, public sports
        {
protected:
        float total;
public:
        void display (void);
};
        void result::display (void)
        {
        total=part1+part2+score;
        put_number( );
        put_marks( );
        put_score( );
        cout<<"\n Total="<<total;
        }
        void main( )
        {
        result R;
        R.get_number (127);
        R.get_marks (83.0, 83.0);
        R.get_score (83.0);
        R.display( );
        }
```

VIRTUAL FUNCTIONS AND POLYMORPHISM

Q. 79 What is polymorphism ? Explain runtime and compile time polymorphism.
(Oct. 2006, Mar. 2009, July 2017)

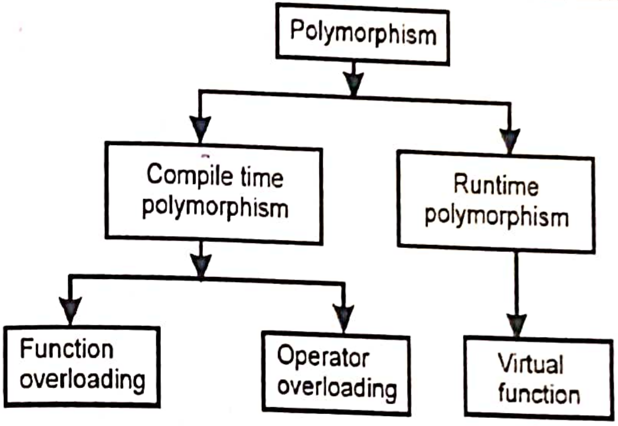
What does polymorphism in C++ ? How is the same achieved at –
(i) Compile time (ii) Runtime ? (Oct. 2002,03,05,14 ; Mar. 2006,13,16, 20)

Ans. :

1) "Polymorphism refers to identically named methods (member functions) that have different behaviour depending on the type of object they refer."

2) Polymorphism simply means "one name, multiple forms."

3) The types of polymorphisms and their examples are shown in following figure.



I) Compile time polymorphism :

1) Function overloading and operator overloading are the examples of compile time polymorphism.

2) In this case, the overloaded member functions are selected for invoking by matching arguments, both type and number.

3) This information is known to the compiler at the compile time and, therefore the compiler is able to select the appropriate function for a particular call at the compile time itself. This is known as compile time polymorphism.

4) Compile time polymorphism is also called as early binding or static binding or static linking. Early binding simply means that an object is bound to its function at compile time.

II) Runtime polymorphism :

1) In some situations, it is nice to select appropriate member function to be invoked while the program is running. This is known as runtime polymorphism.

2) e.g. consider a situation where the function name and prototype is the same in both the base and derived classes as shown in following class definitions.

```
class A
{
        int x;              // private by default.
public:
        void show (void)    // show( ) in base class
        { ... }
    };
        class B: public A
    {
    int y;
    public:
        void show (void)    // show( ) in derived class
        { ... }
    };
```

Here, show( ) function is used to print values of object of both the classes A and B. The prototype of show( ) is the same in both the places, the function is not overloaded and therefore static binding does not apply.

3) In such situations, the appropriate member function can be selected at runtime and it is known as runtime polymorphism.

4) To achieve runtime polymorphism, C++ supports mechanism of virtual functions.

5) At runtime, it is known what class objects are under consideration, the appropriate version of function is called.

6) Since the function is linked with a particular class much after its compilation, this process is termed as late binding. It is also called as dynamic binding because the selection of the appropriate function is done dynamically at runtime.

(July 2019)

Q. 80 Explain the concept of virtual functions.

Ans. :

1) When user use the same function name in both the base and derived classes, the function in base class is declared as virtual using the keyword 'virtual' preceding its normal declaration.

2) When a function is made virtual, C++ determines which function to use at runtime, based on the type of object pointed to by the base pointer.

3) Thus, by making the base pointer to point two different objects, it can execute different versions of the virtual function.

4) Virtual functions can be accessed through the use of a pointer declared as a pointer to the base class.

5) Also, the prototypes of the base class version of a virtual function and all the derived class versions must be identical.

6) If two functions with the same name have different prototype, C++ considers them as overloaded functions, and the virtual function mechanism is ignored.

Q. 81 Explain the difference between static and dynamic binding with example.

(March 2003)

Ans. :

1) *In static binding, object is bound to its function call at compile time.*
While in dynamic binding, selection of the appropriate function is done dynamically at runtime.

2) In static binding, compiler knows the function information (argument type, number etc) at the compile time itself so as able to select appropriate function for a particular call (also called early binding). In dynamic binding, function is linked with a particular class much later after the compilation (also known as late binding).

3) Function overloading and operator overloading are the examples of static binding.

4) Virtual functions are used to implement dynamic binding.

For example :

Consider the following class definitions :

```
class A
{       int x;
        public :
        void show ( )
        {       cout << "Base class";
        }
};
class B : public A
{       int y;
        public :
                void show ( )
                { cout << "'Derived class";
                }
};
```

In above example, compiler does not know which show ( ) function is executed either of base class or derived class. So compiler defers this decision and at the run time select appropriate function concept of virtual function.

In dynamic binding, classes are defined as :

```
class A
{       int x;
        public :
        virtual void show ( )
        { cout << "Base class"; }
};
class B : public A
{       int y;
        public :
        void show ( )
        {       cout << "'Derived class";}
};
```

Q. 82 State any eight basic rules for virtual functions that satisfy the compiler requirements. **(Marh 2002,07,14 ; Oct. 2005,07,13, March 2018; July 2018, 19)**

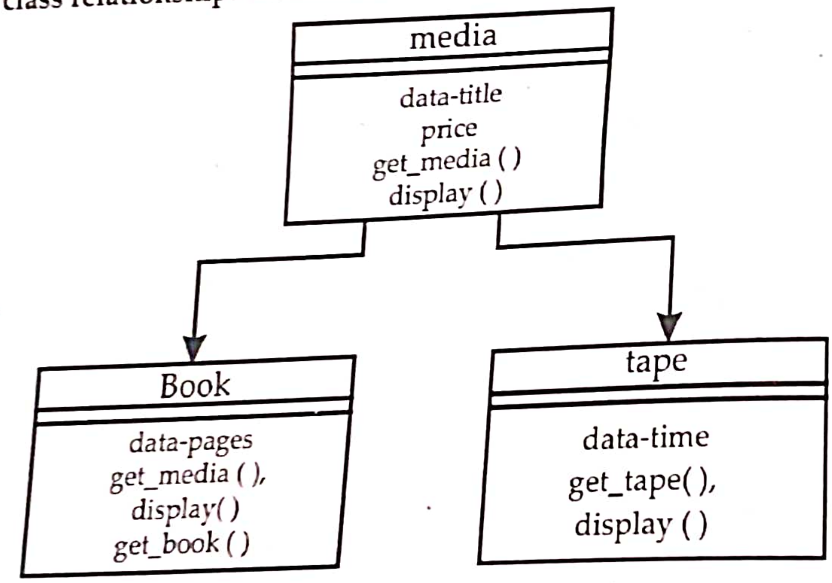
Ans. : When virtual functions are created for implementing late binding, we should observe following basic rules that satisfy the compiler requirements :

1) The virtual functions must be members of some class.
2) They cannot be static members.
3) They are accessed by using object pointers.
4) A virtual function can be a friend of another class.
5) A virtual function in a base class must be defined, eventhough it is not used.
6) The prototypes of the base class version of virtual function and all derived class version must be identical. If two functions have different prototypes, then C++ considers them as overloaded functions and not as virtual functions.
7) We cannot have virtual constructors, but we can have virtual destructors.
8) A base pointer can point to any type of derived object, the reverse is not true. i.e. we cannot use a pointer to derived class to access an object of the base type.
9) When base pointer points to derived class, the incrementation and decrementation is only relative to its base type.
10) Virtual functions are defined in base class, they need not be redefined in derived class.

Q. 83 Write a program to declare a class media, which contains title and price. Declare another two classes book and tape with base class media, which contains pages and time respectively.

Read book details and display it.

The class relationships and functions are shown below :



and You may use additional data & functions. Use virtual functions.

Ans. : *// Runtime Polymorphism*

```
        #include <iostream.h>
        #include <conio.h>
        class media
        {
        protected:
        char title [20];
        float price;
```

```
public:
void get_media (void);
virtual void display (void);
};
class book : public media
{
protected:
        int pages;
public:
void get_book (void);
void display (void);
};
class tape: public media
{
protected:
        int time;
public:
void get_tape (void);
void display (void);
};
void media :: get_media (void)
{
cout<<"Enter title and price" ;
cin>> title;
cin>> price;
}
void media:: display (void)
{
cout<<"\n Title : << title;
cout<<" \n Price : "<< price;
}
void book :: get_book (void)
{
cout<<"\n Enter number of pages";
cin>> pages;
}
void book :: display (void)
{
cout<<"\n Pages" <<pages;
}
void tape :: get_tape (void)
```

```
{
        cout<< "\n Enter time";
        cin>> time;
}
void tape :: display (void)
{
        cout<<"\n Time" <<time;
}
void main( )
{
media m,*p;
book B;
tape T;
m.get_media( );
B.get_book( );
T.get_tape( );
clrscr( );
p = &m;
p → display( )
p = &B;
p → display( );
p = &T;
p → display( );
}
```

Q. 84 **Create a base class shape. Use this class to store two double type values that c be used to compute the area of figures. Derive two specific classes called tria and rectangle from the base shape. Add to the base class a member fun get_data() to initialize base class data members and another member fun display_area( ) to compute and display area of figures. Make display_area( ) virtual function and redefine this function in the derived classes to suit requirements.**

Using these three classes design a program that will accept dimensions of a tria or rectangle interactively and display the area.

Remember the two values gives as input will be treated as lengths of two sid case of rectangle and as base and height in case of triangle. Use the follo formulae,

Area of rectangle = x * y

Area of triangle = 1/2 * x * y

Ans. : // C++ program using virtual function

```
#include <iostream.h>
#include <conio.h>
class shape
{
```

```
protected:
        double x,y;
public:
        void get_data(void);
virtual void display_area(void) // Empty virtual function
{       }
};
class triangle : public shape
{
protected:
        double at;
public:
        void display_area (void);
};
class rectangle: public shape
{
protected:
        double ar;
public:
        void display_area (void);
};
void shape :: get_data (void)
{
        cout <<"\n Enter base and height";
        cin>> x >> y;
}
void triangle :: display_area (void)
{
        at = (1/2)*x*y;
        cout"<<"\n Area of triangle is" << at;
}
void rectangle :: display_area (void)
{
        ar = x*y;
        cout<< "\n Area of rectangle is :" << ar;
}
void main( )
{
        shape s, *P;
        triangle t;
```

```
rectangle r;
s.get_data( );
Γ = &t;
Γ → display_area( );
Γ = &r;
Γ → display_area( );
}
```

WORKING WITH FILES

Q. 85 What are input and output streams ? **(March 2006, 2**

Ans. :

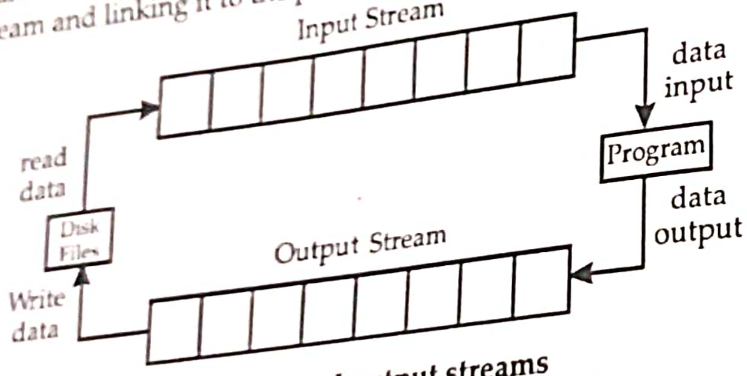
1) The I/O system of C++ handles file operations which are very much similar to con input and output operations.

2) It uses file streams as an interface between the programs and the files.

3) The streams that supply data to the program is known as input stream, while the str that receives data from the program is known as output stream.

4) In other words, input stream extracts (or reads) data from the file and the output str inserts (or writes) data to the file. This is illustrated in following Figure.

5) The input operation involves the creation of an input stream & linking it with program and the input file. Similarly, the output operation involves establishing output stream and linking it to the program and the output file.



File input and output streams

Q. 86 Describe the various classes available for file operations. **(Oct. 2**

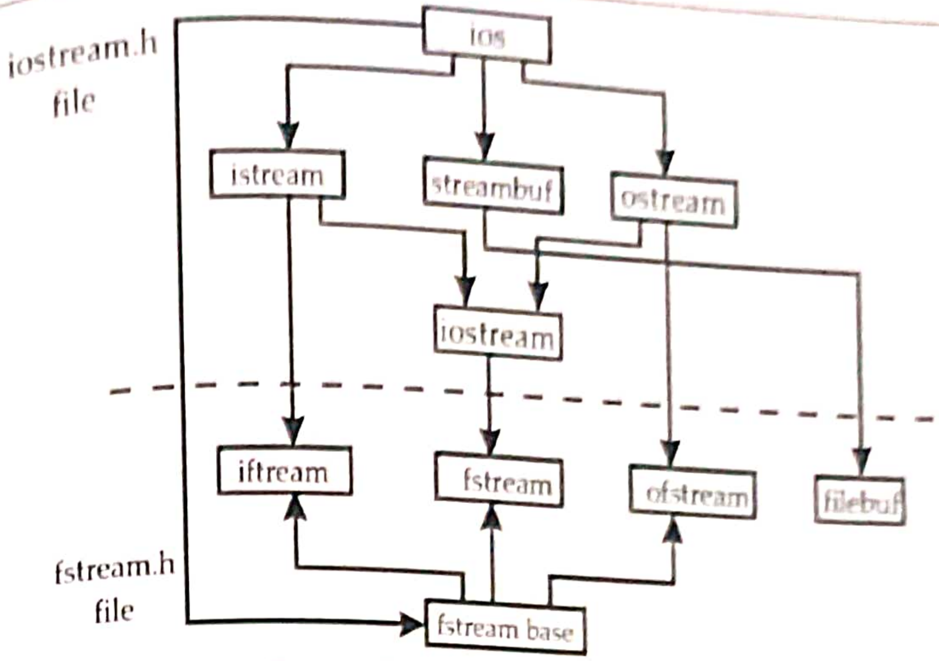
OR Describe briefly the features of I/O system supported by C++ with suita example. **(Oct. 2**

Ans. :

1) The I/O system of C++ contains a set of classes that defines the file handling meth These include ifstream, ofstream and fstream.

2) These classes are derived from fstreambase and the corresponding iostream. h clas shown in the following figure.

3) These classes designed to manage the disk files, are declared in fstream. h and theref we must include this file in any program that uses files.



Stream classes for file operations

The details of file stream classes is given in the following table.

Class	Contents
1. filebuf	Its purpose is to set the file buffers to read and write. Contains 'openprot' constant and used in the 'open( )' of file stream classes. Also contains close ( ) and open( ) as members.
2. fstreambase	Provides operations common to the file streams. Serves as a base for fstream, ofstream, & ifstream classes. Contains open( ) & close( ) functions.
3. ifstream **(March 2020)**	Provides input operations and Contains open( ) with default input mode. Inherits the functions get( ), getline( ), read( ), seekg( ), and tellg( ) from istream.
4. ofstream **(March 2020)**	Provides output operations. Contains open( ) with default output mode. Inherits put( ), seekp( ), tellp( ) and write( ) functions from ostream.
5. fstream **(March 2020)**	Provides support for simultaneous input and output operations. Contains open( ) with default input mode. Inherits all the functions from istream and ostream classes through iostream.

Q. 87 What is the function of each of each of the following file stream classes ?
(i) ifstream (ii) ofstream (iii) filebuf **(Mar. 08 ; Oct. 2003,12)**

Ans. :

(i) **ifstream :** Provides input operations. This file stream class is used to read a stream of objects from a file. **(March 2020)**

(ii) **ofstream :** Provides output operations. Ofstream class is used to write a stream of objects in a file. **(March 2020)**

(iii) **filebuf :** Its purpose is to set the file buffers to read and write. It contains close( ) and open ( ) as members.

Q. 88 State the details of the following file stream classes :
(i) ifstream (ii) ofstream

C++ **(March 2005)**

Ans. :

(a) ifstream :

(1) This class is used for file handling methods.

(2) This class is designed to manage the disk files and user must include this file in any program that uses files.

(3) It provides input operations.

(4) It contains open ( ) function wit default input mode.

(5) It inherits get( ), getline( ), read( ), seekg( ) and tellg( ) functions from istream.

(b) ofstream :

(1) This class is also used for file handling methods.

(2) It provides output operations related to files.

(3) It contains open( ) function with default output mode.

(4) It inherits put( ), seekp( ), tellp( ) and write( ) function from ostream.

Q. 89 How we can open a file using open() function ? **(July 2016, 1**

Ans. :

We can open files using open( ) by two ways :

1) In first method, open( ) function takes only one argument and that is file name.

This is done as follows :

```
file-stream-class stream_object;
stream_object.open("file name");
```

e.g. ofstream fout;

fout.open("Try");

2) A file can be also opened using open() by passing two arguments. The first argument is file name and the second argument is used to specify the file mode. The general form of open() with two arguments is,

```
stream_object.open("file name", mode);
```

The second argument mode specifies the purpose for which file is opened.

e.g. fout.open("computer", ios::app);

This opens the file in append mode.

Note : By using open( ), we can open a file explicitly, whereas, we can also open a file implicitly as.

```
file-stream-class-stream-object ("file name");
```

e.g. ofstream outf ("computer")

It will open file computer in output mode

Q. 90 What are different file modes ? **(Oct. 2002,11)**

Ans. : The following table lists the file mode parameters and their meanings.

	Parameter	Meaning
1.	ios::app	Append to end-of-file.
2.	ios::ate	Go to end-of-file on opening
3.	ios::binary	Binary file
4.	ios::in	Open a file for reading only
5.	ios::nocreate	Open fails if the file does not exist.
6.	ios::noreplace	Open fails if the file already exist.
7.	ios::out	Open file for writing only.
8.	ios::trunc	Delete contents of file, if it exist.

Q. 91 What are classes in C++ for file stream operation ? How do you open and close file in C++ ? Explain any four file modes. **(March 2004,12,16, July 2016)**

Ans. : Classes for file stream operation :

(1) The I/O system of C++ contains a set of classes that define the file handling methods. They include :

(a) ifstream (b) ofstream

(c) fstream **(Oct. 2012)**

(2) These classes are derived from **fstreambase** and the corresponding **iostream.h** class.

(3) They are defined to manage the disk files and declared **in fstream.h**.

Opening and closing a file :

(1) The general format for opening a file is as :

file-stream-class stream-object;

stream-object.open ("filename");

(2) Here **ifstream** class is used to read a stream of objects from a file and ofstream class is used to write a stream of objects in a file.

(3) For example :

Open a file to read stream of object from 'data'.

ifstream infile;

infile.open ("data");

Open a file to write stream of object from 'data'.

ofstream outfile;

outfile.open ("data");

(4) Closing a file : Function **close ( )** is used to close a file, which is opened for read, write or read and write operations. **(July 2016)**

For example : infile.close( );

File modes :

With class **fstream**, file mode can be specified. The form of open( ) function as Stream-object.open ("file name", mode)

File mode parameters are as follows :

1. ios::app - Append to end-of-file.
2. ios::ate - Go to end-of-file on opening.
3. ios::binary - Binary file.
4. ios::in - Open a file for reading only.

Q. 92 What are file pointers ?

Ans. :

1) "Each file has two associated pointers known as the file pointers". One of them is calle the input pointer and the other is called the output pointer.

2) These pointers are used to move through the files while reading or writing.

3) The input pointer is used for reading the contents of a given file location and the outp pointer is used for writing to a given file location.

4) Each time an input or output operation takes place, the appropriate pointer i automatically advanced.

5) Input pointer is also called as get pointer and output pointer is also called as put pointe

Q. 93 Explain the purpose of following functions with example :
1) seekg( ) 2) seekp( ) 3) tellg( ) 4) tellp( ) **(March 2018; July 201**

Ans. :

1) **Seekg( ) :** This function is used to move the file pointer forwards with given number o bytes. It has the form

```
seekg (unsigned int);
```

e.g.
ifstream inf("xyz.dat");
inf.seekg(10);

In above example, seekg( ) moves input pointer 10 bytes forward.

seekg is associated with **input pointer or get pointer.**

2) ***Seekp( ) :*** *This function is used to reposition file pointer to a given number of bytes. This function is associated with output pointer. Corresponding class to process this function is "ofstream" class.*

It has the form :-

```
seekp (unsigned int);
```

e.g. ofstream outf("xyz.dat");
outf.seekp(10)

In above example, output pointer will point to 10th byte in file, after execution of outf.seekp(10);

3) **tellg( ) :** This function is used to return current file pointer position. This function is associated with input file stream. **(July 2018)**

e.g.
```
ifstream inf;
inf.open("xyz.dat");
int n;
n=inf.tellg( );
```

In above example, tellg( ) will return value zero, because initially, input pointer points to zeroth location.

4) **tellp( ) :** This function is used to return current file pointer (output pointer) position. It is associated with **output file stream.**

e.g.
```
ofstream outf;
outf.open("xyz",ios::app);
int n=outf.tellp();
```

In above example file is opened in append mode, therefore file pointer points to end-of-file character. Hence, tellp( ) returns number of characters present in file xyz.

Note : "Seek" functions can also be used with two arguments as :

```
seekg(offset, ref position);
seekp(offset, ref position);
```

Offset represents number of bytes file pointer is to be moved from the location specified by ref position.

The ref position can be one of the three constants :

(i) ios::beg - start of file

(ii) ios::cur - current position of pointer

(iii)i os::end - end-of-file.

Q. 94 Explain the purpose of following functions with example :
i) put( ) ii) get( ) iii) read( ) iv) write( )

Ans. :

(i) **put( ) :** **(July 2016)**

This function is used to store a single character into file, specified by object of ofstream.

It has the form :

```
ofstream object.put(character variable);
```

e.g.
```
ofstream outf;
outf.open("xyz");
char c='A';
outf.put(c);
```

(ii) **get( ) :** **(July 2016)**

This function is used to read a character from a file specified by ifstream object.

It has the form

```
ifstream object.get(character variable);
```

e.g.

```
char c;
ifstream inf("xyz");
while(inf.eof( )==0)
//It will read characters from file till it reaches end of file i.e. eof()
{
inf.get(c);
        cout<<c;
}
```

(iii) write( ) :

It has the form

```
write((char*) & variable, sizeof (variable));
```

This function is used with **object of ofstream** and it is used **store data** into file in binary mode i.e. general user cannot read data of file by using 'type' command. Here all the variables of different type are first of all converted into (char*) i.e. pointer to character and second parameter is number of bytes required to store given variable.

e.g.

```
struct s
{
char n[20];
int t;
};
main( )
{
struct s m;
ofstream outf("xyz");
cin>>m.n;
cin>>m.t;
outf.write((char*) & m, sizeof (m));
}
```

(iv) read( ) :

It has the form

```
read((char*)& variable,sizeof (variable));
```

Data which is stored by using write( ) can be read by using read( ). This function is used to **read data in binary mode** from file. This function is associated with **ifstream object.**

Generally read( ) and write( ) functions are used with structures or objects to store the records. These functions has two arguments. First is address of variable of any data type. But, data is converted into pointer to characters and the second parameter is size of that variable.

e.g.

```
struct s
{
char n[20];
int t;
};
        main( )

{
ifstream inf("xyz");
struct s m;
while (inf.eof( )==0)
{
        read((char*) & m, sizeof (m));
        cout<<"\n"<<m.n;
        cout<<"\n"<<m.t;

}
```

Q. 95 Write a program in C++ to read the name of country from one text file and name of corresponding capital from another text file. The program must display the country name and corresponding capitals name in the output.

Ans. : //Working with multiple files

```
#include<fstream.h>
#include<conio.h>
void main( )
{
        clrscr( );
        ofstream fout;
        fout.open("country");
        fout<<"United States of America \n";
        fout<<"United Kingdom \n";
        fout<<"South Korea \n";
        fout.close( );
        fout.open("capital");
        fout<<"Washington \n";
        fout<<"London \n";
        fout<<"Seoul \n";
        fout.close( );
        const int n=80;
        char line[n];
        ifstream fin;
```

```
                fin.open("country");
                cout<<"Contents of country file: \n";
                while(fin.eof()==0))
                {
                fin.getline(line, n);
                cout<<line;
                }
                fin.close( );
                fin.open("capital");
                cout<<"\n Contents of capital file:\n";
                while (fin) //or equivalent to while (fin.eof)()==0))
                {
                fin.getline(line, n);
                        cout<<"line";
                }
                fin.close( );
        }
```

The output of above program will be

```
Contents of country file :
    United States of America
    United Kingdom
    South Korea
Contents of capital file :
    Washington
    London
    Seoul
```

Q. 96 Write a program to store records of 10 students into file student.dat. Declare class student with member variables name, roll number, marks of three subjects and total marks. By using object of this class and read( ), write( ) functions store records in file student.dat and display those records whose total is greater than 250.

(Note : It is not necessary to use array of objects.)

Ans. : *//C++ program that working with files*

```
        #include<conio.h>
        #include<iostream.h>
        #include<fstream.h>
        class student
        {
        public:
                char name[30];
                int roll_no, m1, m2, m3, total;
```

```
          void getdata (void);
};
void student::getdata (void)
{
cout<<“\n Enter name of student:”;
cin>>name;
cout<<“\n Enter roll number:”;
cin>>roll_no;
cout<<“\n Enter marks in three subjects:”;
cin>>m1>>m2>>m3;
total = m1 + m2 + m3;
}
void main( )
{
          ofstream outf;
          student S;
          int i;
          outf.open(“student.dat”);
          for (i=0; i<=9; i++)
          {
          S.getdata( );
          outf.write((char*)&S, sizeof (S));
          }
          outf.close( );
          ifstream inf;
          inf.open(“student.dat”);
          while (inf)
          {
          inf.read((char*)&S, sizeof (S));
          if (total>250)
          {
          cout<<“Name:”<<name<<endl;
          cout<<“Roll number:”<<roll-no<<endl;
          cout<<“M1=”<<m1<<“\t”<<“M2=”<<m2;
          cout<<“\t M3=”<<m3;
          cout<<“\n Total=”<<total<<endl;
          }
          }
          inf.close( );
}
```

Q. 97 Write a program in C++ to store records of 10 students in file student.dat using object of class student. Each record contains name, roll number and total marks. Modify the record, if total is less than 50.

Ans. : //C++ program using files that modify the record

```
#include<iostream.h>
#include<fstream.h>
class stud
{
public:
        char name[30];
        int roll_no, total;
        void getdata (void);
 };
 void stud::getdata (void)
 {
 cout<<"\n Enter students name:";
 cin>>name;
 cout<<"\n Enter roll number:";
 cin>>roll_no;
 cout<<"\n Enter total";
 cin>>total;
 }
 void main( )
 {
         stud S;
         ofstream outf;
         outf.open("student.dat");
         for(int i=0; i<=9; i++)
         {
         S.getdata( );
         outf.write ((char*)&S, sizeof(S));
         }
         outf.close( ),
         ifstream inf;
         inf.open("student.dat", ios::in | ios::out);
         while (inf)
         {
         inf.read((char*)&S, sizeof (S));
         if (S.total<50)
{
         S.getdata( );
```

```
            inf.seekg(-(size of (S)), ios::cur);
            inf.write((char*)&S, sizeof(S));
    }
            inf.close( );
}
```

Q.98 Write a program in C++ that accepts a string. Store all the characters of string in file "Text", using put(). Use get() to read all the characters of string one by one and display them.

Ans.: //C++ program to read a file and display its contents

```
#include<fstream.h>
#include<iostream.h>
#include<string.h>
void main( )
{
char string[80];
cout<<"\n Enter a string:"<<endl;
cin>>string;
int len;
len=strlen(string);
fstream file;            //input and output stream
file.open("Text", ios::in | ios::out);
for (int i=0; i<=len; i++)
{
file.put(string [i]);    //put a character to file
}
file.seekg(0);           //go to the start
char ch;
while (file)
{
file.get(ch);            //get a character from file
cout<<ch;                //display it on screen
}
}
```

Q.99 Write a declaration for each of the following :

(a) An array a of six doubles. (b) An array a of six pointers to double.

(c) A pointer a to an array of 6 doubles. **(March 2002)**

Ans. :

(a) double a[6];

(b) double *a[6];

(c) double b[6], *a; a=&b[o]; (or a=&b;)

Q. 100 Write declaration for each of the following in C++.

i) An array of 8 floats.

ii) A pointer to an array of 8 doubles.

iii) Function that return a pointer to float.

(Oct. 2007,

Ans : i) float rev [8];

ii) double data [8], i.e. data [8], *a * a ; a = & data [0] ; (or a = & data)

iii) float * fun ( );

Q. 101 Write a declaration for each of the following :

(i) A pointer to an array of 8 floats.

(ii) A function that returns a float

(iii) An array of 8 pointers to float

(Oct. 2008,

Ans. i) float a[8], *ptr; ptr=&a[0]; ii) float fun (); iii) float *a[8];

Q. 102 Write declarations of the following : **(March 2014,**

(i) An array of 8 **(ii) An array of 8 pointers to floats**

(iii) Prototype of function that returns pointer to float (no parameters).

Ans. : (i) float a[8] ; (ii) float *a[8] ; (iii) float *f()

Q. 102(A) Write C++ declaration for the following. **(March 2020,**

(i) Array of 10 integers. , (ii) Pointer to character variable

(iii) Object of the class

Ans. : (i) int a[10]; (ii) char* p; (iii) test t;

Q. 103 Write a C++ program that right justifies text. It should read and echo a sequence o left justified lines and print them in right justified format.

(March 05, 06, 07 ; Oct. 03, 0

Ans. : //C++ program that right justifies text.

```
#include <iostream.h>
int main ( )
{
    const int SIZE = 100; // max. no. of lines stored
    string line [SIZE], S;
    int n = 0, len, maxlen = 0;
    while (! cin.eof ( ))
    {
        getline (cin, S);
        len = S.length ( );
        if (len > 0)
            cout << S << endl;
        if (len > maxlen)
            maxlen = len;
        line [n++] = S;
    }
```

```
                --n;          // n = number of lines read
                for (int i = 0; i < n; i++)
                {
                        S = line [i];
                        len = S.length ( );
                        cout << string (maxlen - len, ' ' ) << S << endl;
                }
        }
```

Q. 104 Implement a circle class. Each object of this class will represent a circle, accepting its radius value as float. Include an area ( ) function which will calculate the area of circle. **(Oct. 2003, 05 ; March 2015)**

Ans. :

```
//C++ program to implement a circle class
    #include <iostream.h>
    class circle
    {
                float a;
                float r;
          public :
                void area (void);
    };
    void circle : : area (void)
    {
          cout << "Enter radius of circle";
          cin >> r;
          a = 3.142 *r*r;
          cout << "The area of a circle is";
          cout << a;
    }
    void main ( )
    {
          circle C;
          C.area ( );
    }
```

Q. 105 Write a function that uses pointer to search for the address of a given integer of a given array. If the given integer is found, the function returns its address, otherwise it returns NULL. **(October 2003, March 2005)**

Ans. :

```
    #include <iostream.h>
          int *location (int a[ ], int n, int target)
          {
                for (int i = 0; i < n; i++)
```

```
        if (a[
            return (
        return NULL;

    }
    int main ()
    {
    int a[8] = {1, 3, 7, 9, 10, 12, 17, 25};
    int *p, n;
    do
        {
            cout <<"Enter the number";
            cin >> n;
            if (p = location (a, 8, n))
                cout <<"The number" << n <<
                    "is found at location" << p ;

            else
                cout << "The number" << n << "is not found" << end


        }
    while (n > 0);
    }
```

Q. 106 Write a C++ program to accept a number and test whether it is prime or not.
(October 2003, 2008; March 2007, March 20

Ans. : //C++ program to test whether the inputted number is prime or not

```
    #include <iostream.h>
    void main ()
    {
            int prime, C = 0;
            cout << "Enter the number";
            cin >> prime;
            for (int i = 2; i < prime; i++)

        {
            if (prime % i == 0)
                C = 1;

        }
        if (C == 0)
            cout << "The number" << prime
                << "is prime number";

        else
            cout << "The number" << prime
                << "is not a prime number";

}
```

Q.107 Write a C++ program to replace every space in an inputed string (less than 80 characters) with a hyphen (i.e. -) **(Mar. - 2004,07; Oct.2012)**

Ans. : //C++ program to replace every space in string with a hyphen.

```
#include <iostream.h>
#include <string.h>
int main ( )
          int len;
{
          char str [80];
          cout << "Enter a string";
          cin.get (str, 80);
          len = strlen (str);
          for (int i = 0; i < len; i++)
          if (str [i] == ' ' )
{
          str [i] = '_';

}
cout << "The final string is";
cout << str ;
}
```

Q. 108 Write an object oriented program in C++ to read an integer number and find the sum of digits of integer [Hint : input 125 output 8 i.e. 1 + 2 + 5 = 81] **(March 2018)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
void main()
{
int val, num, sum = 0;
cout « "Enter the number : ";
cin » val;
num = val;
while (num ! = 0)
{
        sum = sum + num % 10;
        num = num / 10;
}
        cout « "The sum of the digits of" << val << "is" << sum;
}
```

Q. 109 Write a C++ program to accept a sentence and print sentence using pointer. **(July 2018)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
```

Q.110 **Write the output of the following C++ program :** (Mar. 2

Ans. :

```
        #include <iostream.h>
        long comb (int n, int k);
        int main ( )
        {
             const int m = 5;
        for (int i = 0; i < m; i++)
         {
             for (int j = 1; j < m - i; j++)
                    cout << setw (2) << " ";
             for (int j = 0; j <= i ; j++)
                    cout << setw (4) << comb (i, j);
             cout << endl;
         }
      }
      long comb (int n, int k)
      {
           if (n < 0 | | k < 0 | | k > n) return 0;
           long c = 1;
           for (int i = 1; i <=k; i++, n - - )
                  c = c * n/i;
           return c;
    }
```

Ans. : Output of the program is as follows :

```
            1
          1   1
        1   2   1
      1   3   3   1
    1   4   6   4   1
```

Q.111 Write a C++ Program to exchange the contents of two variables using call by reference. **(Oct.04)**

Ans. : // Program to exchange the contents of two variables using call by // reference

```
# include <iostream.h>
void swap (int *, int *); // function prototype
void main ( )
{
int a, b;
cout << "Enter the values";
cin >> a >> b;
cout << "Before Swapping";
cout << "a =" << a;
cout << "b=" << b;
swap (& a, & b);        // call by reference
cout << "After Swapping";
cout << "a =" << a;
cout << "b =" << b;
}
void swap (int *a, int *b)    // function definition
{
int temp;
temp = *a ;   // assign the value at address a to temp
*a = *b;      // put the value at b into a
*b = temp;    // put the value at temp into b
}.
```

Q.112 Write a program in C++ to read a set of numbers from keyboard and find out the largest number in the given array. **(Oct. 2004)**

Ans. : // Program to find out largest number from the given array

```
# include <iostream.h>
void main ( )
{
        int num [ 10 ], max ;
        cout << "Enter the number";
        for (int i = 0; i < 10; i++)
```

```
          cin >> num [ i ];
          max = num [ 0 ];
          for (int j = 1; j < 10; j++)
      {
          if (max < num [ j ])
          max = num [ j ];
      }
          cout << "The largest number in the array is" << max;
      }
```

Q.113 Write a program in C++ to find Greatest Common Divisor (GCD) of two natural numbers. **(Oct. 2004, March 2006, 2011, July 2017)**

Ans. : // To find Greatest Common Divisor of two natural numbers.

```
# include <iostream.h>
void main ( )
{
int n1, n2;
cout << "Enter the two natural numbers";
cin >> n1 >> n2;
while (n1 ! = n2)
{
if (n1 > n2)
n1 = n1 - n2;
if (n2 > n1)
n2 = n2 - n1;
}
cout << "The GCD is :" << n1;
}
```

Q.114 Write a program in C++ that inputs and stores 10 numbers in an array and prints the sum and average of the array elements. **(Mar.2005, 09, 20 ; July 2019)**

Ans. : // Program to print the sum and average of the array elements.

```
# include <iostream.h>
void main ( )
{
int num [ 10 ], sum ;
float avg = 0.0;
cout <<"Enter the 10 elements";
for (int i = 1; i < = 10; i++)
cin >> num [ i ];
sum = 0;
for (i = 1; i < = 10; i ++)
{
sum = sum + num [ i ];
}
avg = sum / 10;
cout <<"The sum of numbers is :" <<sum << endl;
cout << "The average of the array element is : "<< avg;
}
```

Q.115 Write a C++ program to find the smallest of four given integers using min ( ) function to return the smallest of four given integers. int min (int, int, int, int) **(March 2005,13, 18,19)**

Ans. : //Program to find the smallest of four given integers using function.

```
# include <iostream.h>
void main ()
{
        int a, b, c, d, small ;
        int min (int, int, int, int); //Prototype
        cout << "Enter the four numbers :" <<endl;
        cin >> a >> b >> c >> d;
        small = min (a, b, c, d); //function call
        cout << "The smallest number is :" <<small;
}
// function definition
int min (int n1, int n2, int n3, int n4)
{
        int low;
        if (n1 < n2)
        low = n1;
        else
        low = n2;
        if (n3 < low)
        low = n3;
        if (n4 < low)
        low = n4;
        return (low);
}
```

Q.116 Write a C++ program to accept the string from the user and reverse a string. **(Oct.2005,2007)**

Ans. : //Reverse a string

```
#include <iostream.h>
void main ()
```

```
{
        char str [80], reverse [80];
        int len, i, k;
        cout << "Enter a string :" <<endl;
        cin>>str;
        for (i = 0; str [i]! = '\0'; i++);
        len = i;
        len - -;
        for (k = 0; len >=0; len - -, k++)
        reverse [k] = str [len];
        reverse [k] = '\0';
        cout <<"The reverse string is :";
        cout <<reverse;
}
```

Q.117 Implement a class temperature to convert degree Fahrenheit Value to degree Celsius Value. [Hint : C/5 = F - 32/9 , where C is temperature in degree Celsius and F is temperature in Fahrenheit degree] **(March 2005, 2006, 2009)**

Ans. : //C++ program to convert degree.Fahrenheit value to degree celsius value.

```
    #include <iostream.h>
    class temperature
    {
            float tcel;
            float tfht;
            public:
            void getdata ( );
            void display ( );
    };
    void temperature : : getdata (void)
    {
            cout <<"Enter the degree Fahrenheit";
            cin >> tfht;
    }
            void temperature : : display (void)
    {
            tcel = 5/9 * (tfht - 32);
            cout << "The degree celsius:";
            cout << tcel;
```

```
}
        void main ( )
{
        temperature t1;
        t1.getdata ( );
        t1.display ( );
}
```

Q.118 Write a program in C++ to read a set of numbers from the keyboard and to find out the largest number in the given array. (The numbers are stored in a random order.) **(Oct.2006)**

Ans. :

```
#include<iostream.h>
    void main (void)
        {
            int a[100];
            int i, n, larg;
            cout <<“How many numbers are in the array ?” << endl;
            cin>>n;
            cout << “Enter the elements”<<endl;
            for (i = 0; i <= n - 1; ++i)
            {
                cin>>a[i];
            }
            cout << “contents of the array” << endl;
            for (i = 0; i <= n - 1; ++i)
            {
                cout << a [i] << ‘\t’;
            }
            cout << endl;
            larg = a[0];
            for (i = 1; i <= n - 1; ++i)
            {
                if (larg < a[i])
                larg = a[i];
            }
            cout << “Largest value in the array =” << larg;
        }
```

Q.119 Write a function that uses pointers to copy an array of double. **(Oct.2006)**

```
Ans. : double* copy (double a[], int n)
        {
              double* p = new double [n];
              for (int i = 0; i < n; i++)
                    p [i] = a[i];
              return p;
        }
        void print (double [], int);
        int main ()
           {
                    double a[8] = {22.2, 33.3, 44.4, 55.5, 66.6, 77.7, 88.8, 99.9};
                    print (a, 8);
                    double*b = copy (a, 8);
                    print (a, 8);
                    print (b, 8);
           }
```

Q.120 Write a function that has passed an array of n pointers to floats and returns a pointer to the maximum of a floats. **(Oct.2006)**

```
Ans. : float* max (float* p[ ], int n)
              {
              float* pmax = p[0];
              for (int i = 1; i<n; i++)
              { if (*p[i] > *pmax) pmax = p[i]; }
              return pmax;
              }
              float * max (float *P [ ], int n);
              main ()
              {
              float a[8] = {44.4, 77.7, 22.2, 88.8, 66.6, 33.3, 99.9, 55.5};
              float* p[8];
              for (int i = 0; i<8; i++)
              p[i]=&a[i]; //p[i] points to a[i]
              float* M = max (p, 8);
              cout<<M<<","<<*M<<endl;
              }
```

Q.121 Write a C++ program with ComputeTriangle ( ) function that returns the area a and perimeter p of a traingle with given side lengths x, y and z. **(Oct.2007, July 2016)**

(void ComputeTriangle (float & a, float & p, float x, float y, float z).

```
Ans :   # include <iostream. h>
        # include <math . h>
```

```
void computerTriangle (float & a, float & p, float X, float y, float z) ;
void main ()
{
float a, p, x, y, z;
cout <<“Enter three side values of triangle”;
cin>>x>>y>>z;
computerTriangle (a, p, x, y, z);
cout<<“The area with three side values are ”<<x<<end1<<y<<endl<<z<<“is”
        <<a<<“ and its perimeter is ”<<p<<end1;

getch( );
}
void ComputeTriangle (float & a, float & p, float X, float y, float z)
{
p = x + y + z;
double s = p / 2.0;
a = sqrt (s * (s - x) * (s - y) * (s - z) );
}
```

Q.122 Write a C++ program with ComputeCircle ( ) function that returns the area a and circumference c of a circle with given radius r. **(Oct.2007)**

void ComputerCircle (float & a, float & c, float r). **(July 2016)**

OR

Void Circle (float & a, float & c, float & r) **(Oct. 2010)**

(Note : Give function name as per hint)

Ans :

```
# include <iostream. h>
# include <conio. h>
void ComputeCircle (float & a, float & c, float r);
void main ( )
{
clrscr ( ) ;
float r, a, c ;
cout << “ Enter the value for radius r”;
cin >> r;
ComputeCircle (a, c, r); cout << “ The area with radius ”
<< r << “is”<< a <” and its ciramference is “ << c << endl ;
getch ( );
}
void ComputeCircle (float & a, float & c, float r)
{
a = 3.14 * r * r;
c = 3.14 * 2 * r;
}
```

Q.123 Write C++ program to print the input string in a reverse order using function, which first locates the end of string. Then it swaps the first character with the last character, the second character with the second last character and so on.

(Oct.2007; March 2019)

Ans : //c++ program to reverse the string

```
# include <iostream.h>
# include <stdio.h>
# include <string.h>
void reverse (char str[ ], int);
void main ( )
{
        char str [80];
        cout << "Enter the string";
        gets (str);
        int len = strlen (str);
                reverse (str,len);
}
void reverse (char str1[ ], int l)
{
        int mid = l/2;
        for (int i = 1; i < = mid ; i + +)
        {
                char temp = str1 [i];
                str1 [i] = str1 [l];
                str1 [l] = temp;
                l - - ;
        }
        cout << "Reverse of string is :";
        puts (str1);
}
```

Q.124 Write a C+ + program with ComputeSphere ( ) function that returns the volume V and the Surface area S of a sphere with given radius r.

void ComputeSphere (float & S, float& V, float r) **(Mar.2013, 20; Oct.2007, 2008)**

Ans.

```
#include<iostream.h>
#include<conio.h>
void ComputeSphere (float & s, float & v, float r);
void main ( )
{
float s, v, r;
cout<<"Enter the radius";
cin >> r;
ComputeSphere(s, v, r);
cout<<"The area with radius "<<r<<" is "<<s
      <<"and its volume is "<<v<<endl;

getch( );
}
void ComputeSphere (float & s, float & v, float r)

{
const float PI = 3.142;
s = 4.0 * PI *r *r;
v = s * r /3.0;
}
```

Q.125 Write a C++ program to read 5 elements of int array in reserve order and print the array (i.e. Read A [5] first and while printing, print A[0] first). **(Oct.2008)**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main ( )
{
const int SIZE = 5 ;
double a [SIZE];
cout<<"Enter "<<SIZE <<"numbers : " <<endl;
for ( int i = SIZE - 1; i > = 0 ; i -- )
{
cout << "\t a ["<< i <<"] = ";
cin>> a[i];
}
cout<<"In reverse order they are :"<<endl;
for ( i = 0 ; i < SIZE ; i ++)
cout << " \t a["<<i>>"] = "<< a[i]<<endl;
getch ( );
}
```

Q.126 Write a C++ program with average () function that returns the average of four input numbers. **(Oct.2008, July 2016)**
(double ave (double x1, double x2, double x3, double x4);)

Ans.

```
#include<iostream.h>
#include<conio.h>
double ave (double, double, double, double);
void main ( )
{       double a, b, c, d;
        cout << "Enter four numbers : ";
        cin >> a >> b >> c >> d;
        cout <<"The average of all four is : "
              << ave (a, b, c, d) << endl;
        getch ( );
    }
    double ave (double x1, double x2, double x3, double x4)
    {
        double sum = x1 + x2 + x3 + x4;
        return (sum / 4.0);
    }
```

Q. 127 Write an object oriented program to implement a class convert to convert degree Centigrade values to Fahrenheit degree. (Mar.2009)

(Hint C= 5/9 (F-32), where C is temperature in degree Celcius and F is temperature in Fahrenheit degree)

Ans :
```
// c++ program to convert degree celsius value to degree Fahrenheit
    # include <iostream.h>
    class temperature
    {
        float tcel;
        float tfht;
        public :
            void getdata ( );
            void display ( );
    };
    void temperature : : getdata (void)
    {
        cout <<"Enter the degree celcius:";
        cin >> tcel;
    }
    void temperature : : display ( )
    {
        tfht = 9/5 * (tcel + 32);
        cout << "The Fahrenheit temperature :";
        cout << tfht;
    }
    void main ( )
    {   temperature t;
        t.getdata ( );
        t. display ( ); }
```

Q. 128 Write a program in C++ to find sum of First 100 Natural Numbers. **(October. 2009)**

Ans :

```
#include <iostream.h
void main( )
        {int d, sum = 0;
        d = 1;
        while (d<=100)
                {sum = sum + d;
                d = d + 1;
                }
        cout <<"sum" <<sum;

}
```

Q. 129 Write a C++ program to input a word (max. length 15 characters) from user and print each of its characters on new line in Reverse Order. **(October. 09, March 14, July 17)**

Ans :

```
#include <iostream.h>
# include < string.h>
        void main ( )
{
        char x [16] ; int i ;
        cout<<"Enter a word" l;
        cin>>x;
        l = strlen (x);
        for (i = l ; i> = 0 ; i --)
{
        cout <<x [i] <<endl;
        }
}
```

Q. 130 Write a program in C++ using OOP technique to find AREA of Circle. **(October 2009)**

Ans :

```
#include <iostream.h>
class circle
{
        private;
        int r ;
        float A ;
        public:
void getradius()
{
        cin>>r;
}
        void print( )
{
        A = 3.14 * r * r;
cout<< "Area of circle is"<<A;
}
        };
        void main( )
{
        circle c;
        c.getradius( );
        c.print( );
}
```

Q. 131 Write a C++ Program by using swap function to interchange given two numbers
void swap (int & x, int & y) ;
(Mar.2010)

Ans : //C++ program for interchange values.

```
    # include <iostream.h>
    void swap (int & x, int & y);
    void main ( )
    {
        int a, b ;
        cout << "Enter values for a and b";
        cin >> a >> b;
        swap (a, b);
        cout << "After swapping" <<end l;
        cout << "a = " << a;
        cout << "b = " << b;
    }
    void swap (int & x, int & y)
    {
        int temp = x;
        x = y;
        y = temp;
    }
```

Q. 132 Write a C++ Program to count occurrence of a character 'J' in a given string.
(Mar. 2010 ; July 201

Ans : //C++ program to count occurrence of character 'J'

```
    # include <iostream.h>
    # include <string.h>
    void main ( )
    {
        char string [80];
        int len, count, i;
        count = 0;
        cout <<"Enter a string : "<<endl;
        gets (strlen);
        len = strelen (string);
        for (i=0; i<len; i ++)
   {
        if (string [i] = = 'J')
               count ++;
   }
   cout <<"Occurence of character 'J' in given string is :" <<count;
   }
```

Q. 133 Show output of the following C++ Program : (Mar. 2010)

```
class test
{
     private :
     int i, j;
     public :
     void calculate ( )
     {
          for (i = 1; i < = 5; i ++)
          {
               for (j = 1; j< = i; j++)
               { cout << j % 2 << "\t";
               }
               cout << endl;
               }
     }
};
void main ( )
{
     test a ;
     a. calculate ( );
}
```

Ans : output of given C+ + program is as fallows ;

```
1
1   0
1   0   1
1   0   1   0
1   0   1   0   1
```

Q. 134 Write a C++ program to sort 10 integer numbers in ascending order. (Oct.2010)

Ans :

```
void main ()
{ int n [10], i, j, temp ;
cout << " Enter 10 integer" <<endl;
for ( i = 0; i < = 9; i++)
{ cin>>n[i] ;
}
for (i = 0; i < = 9; i ++)
{
for (j = i; j < = 9; j++)
{ if (n [i] > n[j])
{ temp = n[j];
n[j] = n[i];
    n[j] = temp ;
    }
  }
}
cout << " sorted list is" << endl ;
for (i =0; i < =9, i ++)
{ cout << n [i] << endl ;
  }
}
```

Q. 135 Write a C++ program with a function to find whether the year entered is a Leap year or not. 5

Ans :

```
# include <iostream.h>
void main ()
{ int leap (int) ;
int n ;
cout << " Enter year" << endl ;
cin >> n ;
cout << leap (n)
}
int leap (int n)
{
if (n% 4 == 0 && n% 100!= 0 | | n% 400 ==0)
cout << " Year is Leap Year";
else
cout<<< " Year is not Leap Year";
return n;
}
```

Q. 136 Write an Object Oriented Program in C++ to read a set of 10 numbers and store it as an one dimensional array; again read a number 'd' and check whether the number 'd' is present in the array, if it is so, print out how many times the number 'd' is repeated in the array. **5**

Ans:

```
# include <iostream. h>
class search
{
private :
        int num [10] ;

public :
        void getdata( ) ;
        void find ( ) ;

} ;
void search :: getdata ()
{
        cout << " Enter elements";
        for (int i =0 ; i < 10 ; i ++)
            cin >> num [i];

int n ;
cout < < " Enter number ";
cin >> n;
}
void search : : find ()
{ int count = 0 ;
        for ( i = 0 ; i < = 9 ; i ++)
{
        if ( num [i] == n)
        count ++ ;
}
count << " The number " << n << " is
        present in array " << count << " times" ;
}
void main ( )
{
search s ;
s. getdata ( ) ;
s. find ( ) ;
}
```

Q. 137 Write an Object Oriented Program in C++ to read an integer number and find ou the sum of all the digits of a number.

(For eg. n = 1256, SUM = 1 + 2 + 5 + 6 =14)

Ans : // C++ program to read an integer number & find out the sum of all the digits

```
    # include <iostream.h>
    class digit
    {
        private :
            int num ;
        public :
            void getdata ( ) :
            void sum ( ) ;
    } ;
    void digit : : getdata ( )
    {
        cout << " Enter the number" ;
        cin >> num ;
    }
    void digit : : sum ( )
    {
        int rem, add = 0;
        while ( num > 0)
    {
        rem = num % 10;
        add = add + rem;
        num = num /10;
    }
        cout << " The sum of all digit of number= " << add ;
    }
    void main ( )
    {
        digit S ;
    S.getdata ( ) ;
    S.sum ( ) ;
    }
```

Q. 138 Write a program in C++ to read a set of numbers from keyboard and find out larges number in the given array using pointers. **(Oct. 2011, 5 Marks**

Ans.: find largest number in given array using pointers,

```
void main ( )
{
```

```
int a [100], b, *ptr, d, Max = 0;
cout <<"Enter ten numbers"<<endl;
for (d = 0; d <= 9; d++)
    {
            Cin >> a [d];

    }
    ptr = a; // ptr = &a [0]
for (d = 0; d <= 9; d++)
{
If (* ptr > max)
    {
            Max = * ptr;

    }

}
cout <<"largest Number="<<max;

}
```

Q. 139 Write a program in C++ to convert the given binary number into decimal equivalent using method convert ( ). **(Oct. 2011, 5 Marks)**

Ans. :

```
class binary
{
        private :
                int bin;
        public :
                void getdata ( )
                {
                        cout <<"Enter binary number"<<endl;
                        cin >> bin;
                }
                void convert ( )
                {
                        int digit, dec = 0, i = 0;
                        do
                        {
                                digit = bin%10;
                                dec = dec + digit*pow (2,'i);
                                i++;
                                bin = bin/10;
                        } while (bin! = 0);
                cout << "Decimal equivalent="<<dec;
                }
      };
void main ( )
{
binary a;
a.getdata ( );
a.convert ( );
}
```

Q. 140 Implement a class rectangle. Each object of this class, will :represent a rectangle accepting its length and width as float. Include an area function which will calculate the area of, the rectangle. **(Oct. 2011, 5 Marks)**

Ans. :

```
      Class rectangle
      {
            private :
                  float len, width;
            public :
                  void getdata ( );
                  void area ( );
      };
      void rectangle : : getdata ( )
      {
            Cout<<"Enter length and width"<<endl;
            Cin>>len>>width;
      }
      void rectangle : : area ( )
      {
      cout << "Area of rectangle =" <<len*width;
      }
      void main ( )
      {
            rectangle r
            r.getdata ( );
            r.area ( );
      }
```

Q. 141 Write a program in C1 to find GCD and. LCM of two inputted numbers using methods enter ( ) and compute ( ). **(March 2012, 5 Marks)**

Ans.: Find GCD and LCM of two input number,

```
      Class lcmgcd
      {
        int a, b, x,
    public :
        void enter ( )
        {
                cout<<"Enter two number";
                cin>>a>>b;
                x = a * b;

        }
        void compute ( );

};
void lcmgcd : : compute (void)
{       while (a! = b)
        {
                if (a > b)
                        a = a – b;
                if (b > a)
                        b = b – a;

        }
        l = x/a;
        cout<<"GCD = "<<a;
        cout<<"LCM = "«l;

}
void main ( )
        lcmgcd d;
        d.enter ( );
        d.compute( );
}
```

Q. 142 Implement a point class for two dimensional points (x, y). Include a function dist( ) to return points distance from origin (0, 0) and a display ( ) function to print result. **(March 2012, 5 Marks)**

Ans. :

```
class point
{
private :
        double x, y;
public :
        void getdata ( )
        {

                cin>>x>>y;
```

```
            }
            double dist ( )
            {
                    return sqrt (x*x+y*y);
            }
            void display ( )
            {
                    cout<<"Distance between"<<x<<"and"<<y<<dist ( );
            }
    };
    void main ( )
    {
          point c;
          c.getdata ( );
          c.display ( );
    }
```

Q. 143 Write an Object Oriented Program in C++ that prints the factorial of a given number. **(Oct. 2012, 5 Marks)**

Ans. :

Factorial of given no.

```
#include<iostream.h>
class fact
{       private:
              int n;
        public:
              void getaata();
              void display();
};
void fact : : getdata ()
{
        cout<<"Enter value of n\n";
        cin>>n;
}
void fact :: display( )
{
       for (int i=1,f = 1; i<=n; i++)
       {
            f = f * i;
       }
```

```
    cout <<"Factorial o
         <<"=" <<f<<endl;

}
void main ()
{
    fact obj;
    obj. get data();
    obj. Display();
}
```

Q. 144 Write a C++ program to find largest of four given integers using max ( ) function to return largest of four given integers : int max (int, int, int, int). (March 2013, 5 Marks)

Ans. :

```
Void main( )
{
    int max(int, int, int, int);
    cout << "enter four integer";
    int w, x, y, z;
    cin >> w >> x >> y >> z;
    cout << "minimum =" <<max (w, x, y, z);

}
int max (int n1, int n2, int n3, int n4)
{
    int max = n1;
    if (n2 > max) max = n2;
    if (n3 > max) max = n3;
    if (n4 > max) max = n4;
    return max;
}
```

Q. 145 Write a program in C++ that inputs and stores 10 numbers in any array and prints numbers in reverse order. (March 2013, 5 Marks)

Ans. :

```
void main( )
{
    double a[10];
    cout <<"enter number";
    for (int i = 0 ; i= 10; i++)
    {
        count <<"\t a ["<< i <<"]";
        cin>>a[i];
    }
    cont <<"The array in reverse order is\n";
```

```
        for (int j = 9 ; j > 0 ; j –)
        {
            Cout <<"\t a["<<j<<"]";
            Cout << a[j] << endl;
        }
}
```

Q. 146 Write a C++ Program to find factorial of 1 of 5 numbers. **(Oct. 2013, 5 Marks)**

Ans. :

```
    #include <iostream.h>
    void main ( )
    {
        int fact, i, j;
        fact = 1;
        cout << "Number " << "\t" << "Factorial";
            for (j = 1; j< = 5; j++)
            {
                fact = fact * j;
            }
            cout << i << "\t" << fact <<endl;
    }
```

Q. 147 Write a program in C++ to find sum of contents of an array with the help of a pointer. **(March 2014, 5 Marks)**

Ans. :

```
#include<iostream.h>
void main()
{       int *p = &a[0];
            int i,s = 0;
            for (i = 0; i < 5; i++)
                {
                        S=s+*p;
                        p++;
                }
                cout<<"sum="<<s;
}
```

Q. 148 Implement a class circle. Include a constructor in it which accepts value of radius from user. Include two more functions in it, one of which calculates area and circumference and the other prints answers **(March 2014, 5 Marks)**

Ans. :

```
#include<iostream.h>
class circle
```

```
  private :
        int r ;
        float a, c;
  public :
        circle()
        {
              cout<<"Enter Radius";
              Cin>>r;

        }
        void cal()
        {
              a = 3.14 *r*r;
              c = 2*3.14*r;

        }
        void print()
        {
        cout<<a<<c;
        }
        ~ circle(){}
  };
void main()
{
      circle obj;
      obj.cal();
      obj.print();
}
```

Q. 149 Write a program in C++ display the following output using for loop : **(Oct. 2015)**

```
1
1  2
1  2  3
1  2  3  4
1  2  3  4  5
```

Ans. :

(a)

```
# include <isotream.h>
void main ( )
{
      int i, j;
      for (i =1; i < = 5; i++)
      {
            for (j = 1; j < = i; j++)
            {
              for (j = 1; j <= i; j++)
              cout << j << " ";
              cout << endl;
              }
      }
```

Q. 150 Write a C++ program to find the GCD (Greatest Common Divisor) of two numbers entered by user. (March 2014, 5 Marks)

Ans. :

```
#include<iostream.h>
void main()
{
      int a, b,i;
      count <<"Enter Two Numbers";
      cin >>a>>b;
      int gcd =1, min;
      if (a>b)
      {
            min = a;
      }
      else
            min=b;
      for (i=1, i<=min; i++)
      {
     if (a%i==0&&b%i==0)
           gcd = i;
     }
           cout <<"GCD="<<gcd ;
}
```

Q. 151 Write a program in C++ to initialize the array of 10 integers and find the sum of all the elements of array. (Oct. 2014, 5 Marks)

Ans. :

```
    void main ( )
    {
          int a [10], i, sum;
          cout<< " Enter 10 Elements ";
             for ( i=0; i < 10; i++)
   {
        cin>> a [1];
   }
   sum = 0;
```

```
cout<<"/n The Array = ";
    for ( i=0; i<10; i++ )
    cout<< a[i] << "  ";
{   sum = sum + a[i];

}
cout<<"\n sum of all element = " << sum;
    getch ( );

}
```

Q. 152 Write a program in C++ to accept a line from keyboard and count total no. of blank spaces in a line. The program should print the original string and blank spaces. (Oct. 2014, 5 Marks)

Ans. :

```
void main ( )
    {
        char a[80] ;
        int i, bsp;
        cout<< " Enter a sentence = ";
        gets (a);
        bsp = i = 0;
        while ( a[i]  ! = '\0' )
    {   if ( a[i] = = '  ')
                {       bsp++ ;
                }
    i++;
}
cout<<"/n Total Blank space = "  << bsp;
getch ( );
}
```

Q. 153 Write a program in C++ to accept three numbers from keyboard and find the smallest one and print it. (Oct. 2014, 5 Marks)

Ans. :

```
void main ( )
{   int a, b, c;
    cout<< " Enter three no. = ";
    cin >> a>>b>>c;
    int small = a;
    if ( b < small )
{   small = b;  }
if ( c < small )
{ small = c ; }
cout<<"/n smallest no = " << small;
getch ( );
}
```

Q. 154 **Write a program in C++ to initialize the array to 10 floats and print all the array elements using pointer.** **(Oct. 2014, 5 Marks)**

Ans. :

```
void main ( )
{
float a [10] = { 1.2, 2.3, 4.5, 5.6, 7.8, 8.9, 9.1, 6.7, 10.2, 10.4 }
            float *p;
        int i ;
p = & a [0];
            cout <<"\n Array=" ;
        for  ( i=0;  i<=9;  i++ )
{       cout<< * p    << "  " ;
        p++;
}
        getch ( );
}
```

Q. 155 **Write a program in C++ to accept two integer values in main function, pass them to function great() using call by value and find greater number, function great() should not return any value.** **(March 2015, 5 Marks)**

Ans. :

```
void great( int, int);
      void main( )
      {     int a, b;
            cout<<"Enter Two No=";
            cin >> a>>b;
            great(a,b);
            getch( );
      }
   void great( int p, int q)
   {     if( p>q)
         {     cout<<p<<"is big" ;}
               else
         {     cout<<q<<"is big"; }
}
```

Q. 156 Write a program in C++ to accept three integers from keyboard and find greatest number with using condition control. **(March 2015, 5 Marks)**

Ans. :

```
void main( )
{     int a, b, c;
      cout<<"Enter three no. = ";
      cin >> a>>b>>c;
      int big = a;
      if ( b < big )
{     big = b; }
if ( c < big )
{ big = c ; }
cout<<"\n Greatest no = " << big;
getch( );
}
```

Q. 157 Write a program in C++ to accept a string from keyboard and copy string into another string without using the library function. **(March 2015, 5 Marks)**

Ans.

```
void main( )
{     char a[80], b[80];
      int i;
      i = 0
   cout<< " Enter a String = ";
cin>> a;
while( a[i]! ='\0' )
{     b[i] = a[i];
         i++;
}
b[i] = '\0' ;
cout<<"\n copied string="<<b;
getch( );
}
```

Q. 158 Write a program in C++ to declare the array of 10 floats and find the largest.

Ans. : **(Oct. 2015, 5 Marks, March 2017)**

```
                    }
                    int largest = a[0];
                    for (i = 1; i<=9; i++)
                    {
                            if(a[i] > Largest)
                            {
                                    Largest = a[i];
                            }
                    }
            cout<<"Largest of all 10 values = " << Largest;
    }
```

Q. 159 Write a program in C++ to accept a string from keyboard and find the length of the string without using library function. **(Oct. 2015, 5 Marks)**

Ans. : void main ( )

```
    {
        char S[80];
        cout<<"Enter a string" <<endl;
        cin >> getline (S, 80);
        int i = 0, L = 0;
        while (S[i] ! = '\0')
            {
                i++;
                L++;
            }
        cout<<"length of a string = " <<L;
    }
```

Q. 160 Write a program in C++ to accept two integer number in main ( ) and find sum of those no using function of by address. **(Oct. 2015, 5 Marks)**

Ans. :

(b)

```
int Add (int *, int *);
void main ( )
        {       int N1, N2;
                cout <<"Enter two No";
                cin>>N1>>N2;
                int sum = Add (&N1 & N2);
                cout << "Addition = "<<SUM;
        }
  int Add (int * x, int * y)
        {
                int S = *x + *y
                return S;
        }
```

Q.161 Implement a class temperature. Include a constructor in it which accepts value of temperature from user in degree Celsius. Include two functions in it, one of which calculates its equivalent temperature in degree Fahrenheit and other function prints the answer :

[Formula : $\frac{C}{5} = \frac{F-32}{9}$] **(March 2016, 5 Marks)**

Ans. :

```
#include <iostream.h>
class temperature
{     private :
      float f, t;
      public :
          temperature()
          {
                cout <<"Enter temp. in celcin";
                cin >> t;

          }
          void cal ( )
          {
                f = 9* c/5 + 32;

          }
          void print( )
          {
                cout <<"temp in farenheit"<< f
                ~ temperature( ) {  }

          };
          void main( )
                {        temperature ob;
                         ob . cal();
                         ob.print( );

                }
```

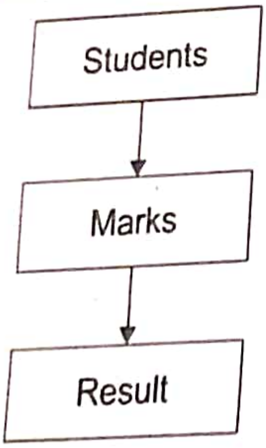
Q.162 Implement a class average. Include a constructor in it which will accept value of three variables from user. Include two more functions in it, one functions calculates average and other prints it. **(March 2016, 5 Marks)**

Ans. :

```
#include <iostream.h>
class average
   {private :
        float a, b, c, s;
    Public :
        average( )
        {
                cout <<"Enter 3 nos." ;
                cin >> a >> b >> c;

        }
    void cal()
        {
                s = (a +b + c)/3;

        }
    void print( )
        {
                cout <<"average=" << s;

        }
        ~ average( ) { }
        };
        void main( )
        { average ob;
         ob.cal( );
         ob.print( );
        }
```

Q. 163 Implement the above class hierarchy of inheritance. Class student accepts roll number of student, class marks accepts marks of three subjects and class result calculates the total and prints all details. (5)

(Create object of class result)



Ans. :

```
# include <iostream.h>
      class student
      { protected : int r ;
      public :
              void get1( )
                 {cout <<"Enter roll no.";
                 cin >>r; }
             };
             class marks : public student
```

```
        { protected : int m1, m2, m3;
        public :
        void get 2( )
        {
            cout <<"Enter marks of 3 sub";
            cin >> m1 >> m2 > m3;

        }
};
class result : public marks
{ protected : int tot ;
public :
        void cal( )
            { tot = m1 + m2 + m3;
            }
        void print ( )
            {
            cout <<"roll no". << r;
            cout <<"total =" << tot; }
            };

void main( )
{           result ob;
            ob.get1( );
            ob.get2( );
            ob.cal( );
            ob.print( );

}
```

Q.164 Impliment class GCD which have member function (a/c), which calculate greatest common divisor of two number entered during program execution. Print() will Print GCD of two number. **(March 2017, July 2019, March 2020)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
class GCD
{
private:
    int a, b;
public:
    void gcd()
    {
        cout<<"\n Enter Two numbers:";
        cin>>a>>b;
        while(a!=b)
        {
            if(a>b)
```

```
                    a=a-b;
            if(b>a)
                    b=b-a;
        }
    }
    void print()
    {
        cout<<"\n GCD="<<a;
    }
};
void main()
{
GCD g;
g.gcd();
g.print();
}
```

Q. 165 Write an object oriented program in CH to read an integer number and find the sum of digits of integer [Hint : input 125 output 8 i.e. 1 + 2 + 5 = 81] **(March 2018)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int val, num, sum = 0;
    cout « "Enter the number : ";
    cin » val;
    num = val;
    while (num ! = 0)
    {
        sum = sum + num % 10;
        num = num / 10;
    }
        cout « "The sum of the digits of" << val << "is" << sum;
}
```

Q. 166 Write a C++ program to accept a sentence and print sentence using pointer. **(July 2018)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
void main()
{
    char a[80],i,*ptr;
    clrscr();
    cout<<"Enter Sentence:";
    cin.gentline(a,80);
    ptr=&a[0];
    cout<<"sentence is:";
    for(i=0;a[i]!='\0';i++)
    {
        cout<<*ptr;
        ptr++;

    }
    getch();
}
```

Q. 167 Write a class based C++ program to print 20 terms of fibonacci series. **(July 2018)**
[Hint : Fibonacci series 0, 1, 1, 2, 3, 5, ...]

Ans. :

Method 1 :

```
#include<iostream.h>
#include<conio.h>
class fibonaaci
{
    private:
    long int f0,f1,fib;
    public:
    fibonacci(void);
    void process(void);
    void display(void);
    void display1(void);
};
fibonacci::fibonacci(void)
{
    f0=0;
    f1=1;
}
void fibonacci::display1(void)
{
    cout<<f0<<"\t"<<f1<<"\t";
}
void fibonacci::process(void)
{
    fib=f0+f1;
```

Method 2 :

```
#include<iostream.h>
#include<conio.h>
class fibonacci
{
        private;
        long inte f0,f1,fib;
        public:
        fibonacci(void);
        void process(void);
};
fibonacci::fibonacci(void)
{
        f0=0;
        f1=1;
}
void fibonacci::process(void)
{
        int i, n;
        cout<<"\n Enter number of
        elements"<<endl;
        cin>>n;
        cout<<f0<<"\t"<<f1<<"\t";
        for(i=3;i<n;i++)
        {
```

```
        f0=f1;
        f1=fib;
    }
    void fibonacci::display(void)
    {
        cout<<fib<<"\t";
    }
    void main()
    {
        int i, n;
        fibonacci f;
        cout<<"\n  Enter  number  of
        elements"<<endl;
        cin>>n;
        f.display1();
        for(i=3;i<=n;i++)
        {
            f.process();
            f.display();
        }
}
```

```
                fib=f0+f1;
                cout<<fib<<"\t";
                f0=f1;
                f1=fib;
        }
}
void main()
{
        clrscr();
        fibonacci f;
        f.process();
        getch();
}
```

Q. 168 Write a C++ program to overload add ( ) function which will add two integer [add(int, int)] and three integers [add (int, int, int)]. **(July-2018)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
int add(int,int);
int add(int,int,int);
void main()
{
    int a,b,p,q,r,a1,a2;
    cout<<"\nEnter two number";
    cin>>a>>b;
    a1=add(a,b);
    cout<<"\n Addition of two numbers="<<a1;
    cout<<"\nEnter three number";
    cin>>p>>q>>r;
    a2=add(p,q,r);
    cout<<"\n Addition of three numbers="<<a2;
    getch();
}
int add(int n1,int n2)
{
    return n1+n2;
}
int add(int m1,int m2,int m3)
{
    return m1+m2+m3;
}
```

Q. 169 Write a C++ program to accept an array of 10 integers and find smallest and largest element in array. **(July 2018)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int a[10],i,j,temp;
    cout<<"Enter 10 integers";
    for(i=0;i<10;i++)
    {
        cin>>a[i];
    }
    for(i=0;i<10;i++)
    {
        for(j=0;j<=10-i;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    cout<<"\nThe smallest number in array is="<<a[0];
    cout<<"\nThe smallest number in array is="<<a[9];
    getch();
}
```

Q. 170 **Write a C++ program to accept a sentence (maximum 50 characters) and print sentence in reverse.** **(March 2019)**

Ans. :

Method 1 :

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main()
{
    char S[50];
    int i, L;
    cout<<"Enter a setnece(50 chars)";
    cin.getline(S,50);
    L=strlen(S);
    cout<<"\n Sentence in reverse";
    for(i=L-1;i>=0;i--)
    {
        cout<<S[i];
    }
    getch();
}
```

Method : 2

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main()
{
    char S[50];
    int i, L;
    cout<<"Enter a sentence(50 chars)";
    cin.getline(S,50);
    cout<<"\n Sentence in reverse";
    strrev(S);
    cout<<S;
}
```

Method 3 : (Prefer This Method)

```
#include<iostream.h>
#include<conio.h>
{
    char S[50];
    int i, L;
    cout<<"Enter a sentence(50 chars)";
    cin.getline(S,50);
    for(i=1;S[i]!='\0';i++)
    {
        L++;
    }
    cout<<"\n Sentence in reverse";
```

Q. 171 Write a C++ program to find smallest in an array of 10 floats using pointer. **(March 2019)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    float a[10],small,*p;
    int i;
    cout<<"\n Enter 10 numbers";
    for (i=0;i<10;i++)
    {
          cin>>a[i];
    }
    p=&a[0];
    small=*p;
    for(i=0;i<10;i++)
    {
          if(*p<small)
          {
                small=*p;
          }
          p++;
    }
    cout<<"\nsmallest element="<<small;
    getch();
}
```

Q. 172 Write a class based program in C++ to find area of a Triangle. **(March 2019)**

Ans. :

```
#include<iostream.h>
#include<conio.h>
class triangle
{
     private:
     float a,b,h;
     public:
     void area();
};
void triangle:: area()
```

```
        {
                cout<<"Enter base and height"<<endl;
                cin>>b>>h;
                a=0.5*b*h;
                cout<<"\nArea of triangle="<<a;
        }
        void main()
        {
                triangle t;
                t.area();
        }
```

Q. 173 Write a class based C++ program to find the area of a sphere. **(July 2019)**

Ans. :

```
    #include<iostream.h>
    #include<conio.h>
    class sphere
    {
        float r, a;
        public:
        void area()
        {
                    cout<<"Enter radius of sphere:";
                    cin>>r;
                    a=4*3.14*r*r;
                    cout<<"area of sphere="<<a;

        }
  };
  void main()
  {
        sphere s;
        s.area();
  }
```

Q. 174 Select the correct alternative and rewrite the following.

1. **float *ptr :**

In above declaration, data type of ptr is —— and data type of variable pointed by ptr is ——

(i) float, float (ii) float, int (iii) int, float (iv) pointer, float

Ans. : (iv) Pointer, float

2. When a function is called by reference, it can work an —— variables in the calling program.
   (i) Original (ii) Virtual (iii) Copies of (iv) None of these

Ans. : (i) Original

3. *ptr ++ means ——
   (i) Increment the content of ptr by size of data type to which ptr is pointer.
   (ii) Increment the content of ptr by 1.
   (iii) Increment the content of memory location pointed by ptr by 1.
   (iv) None of these.

Ans. : (iii) Increment the content of memory location pointed by ptr by 1.

4. Last character of a string is ——
   (i) 0 (ii) \0 (iii) \n (iv) end

Ans. : (ii) 0

5. When we use string functions such as strlen(), strrev() etc. then it must include file ——
   (i) #include<string.h> (ii) #include<iostream.h>
   (iii) #include<fstream.h> (iv) #include<iostring.h>

Ans. : (i) #include<string.h>

6. Objects are basic —— in object oriented programming. (March 2018)
   (i) Run time entities (ii) Compile time entities
   (iii) Data types (iv) None of these.

Ans. : (i) Run time entities

7. Object is a variable, whose data type is ——
   (i) integer (ii) class (iii) structure (iv) float

Ans. : (ii) class

8. —— is not a visibility label. (Oct. 2011)
   (i) Public (ii) Private (iii) Separate (iv) Protected

Ans. : (iii) Separate

9. The members declared under —— visibility label are hidden from external use.
   (i) Public (ii) Private (iii) Both (i) and (ii) (iv) None of (i) and (ii)

Ans. : (ii) Private.

10. If all visibility labels are missing, then by default members of class are ——
    (i) Public (ii) Protected (iii) Private (iv) Any of these (March 2017)

Ans. : (iii) Private

11. When a member function is defined inside the class, then it is treated as a —— function.
    (i) inline (ii) outline (iii) external (iv) virtual

Ans. : (i) inline

12. A class is defined as follows,

```
public:
        friend void getdata (void);

};
```

The correct header for defining getdata() function outside the class is ——

(i) friend void abc::getdata (void)

(ii) void abc::getdata (void)

(iii) friend void getdata (void)

(iv) void getdata (void)

Ans. : (iv) void getdata (void)

13. A constructor can never return a value. Hence it has ....

(i) no return type. (ii) void return type

(iii) int return type (iv) any of these

Ans. : (i) no return type

14. A destructor is invoked implicitly by the compiler upon —— the program.

(i) entry in (ii) exit from

(iii) Mid point of (iv) none of these.

Ans. : (ii) exit from

15. A special function, which is used to define an additional task to an operator is called as ——

(i) operator overloading (ii) operator function

(iii) friend function (iv) constructor

Ans. : (ii) operator function

16. Operator function as a member function will have only one argument for —— operators.

(i) unary (ii) binary (iii) sizeof (iv) none of these

Ans. : (ii) binary

17. —— is the operator which cannot be overloaded. **(Oct. 2002)**

(i) scope resolution (ii) binary (iii) unary (iv) none of these

Ans. : (i) Scope resolution

18. The derivation of one class from another derived class is called as ——

(i) multiple inheritance (ii) single inheritance.

(iii) multilevel inheritance (iv) hybrid inheritance

Ans. : (iii) multilevel inheritance

19. When a class is made —— the compiler takes necessary care to see that only one copy of that class is inherited in derived classes.

(i) virtual base class (ii) base class

(iii) derived class (iv) single class

Ans. : (i) virtual base class

20. In early binding, the correct function to be invoked is selected at ——

(i) runtime (ii) polymorphism time

(iii) compile time (iv) none of these

Ans. : (iii) compile time

21. To achieve run-time polymorphism, C++ supports mechanism of ——

(i) function overloading (ii) operator overloading

(iii) virtual functions (iv) both (i) and (ii)

Ans. : (iii) virtual functions

22. —— stream extracts data from the file.

(i) input (ii) output (iii) input/output (iv) none of these

Ans. : (i) input

23. The class —— is not derived from fstream base class.

(i) filebuf (ii) fstream (iii) ifstream (iv) ofstream.

Ans. : (i) filebuf

24. ios::in means ——

(i) open a file for writing only. (ii) open a file for reading only.

(iii) open fails if file already exists. (iv) open a file in binary mode.

Ans. : (ii) open a file for reading only.

25. C++ was developed by Bjarne stroustrup at ——

(i) Xerox corporation (ii) AT and T Bell laboratories.

(iii) Polo Alu Research Centre (PARC) (iv) None of these.

Ans. : (ii) AT and T Bell laboratories

26. —— is not a built-in datatype in C++.

(i) void (ii) float (iii) char (iv) class

Ans. : (iv) class

27. The do-while statement is ——

(i) an entry control loop (ii) conditional statement

(iii) an exit control loop (iv) none of these.

Ans. : (iii) an exit control loop

28. Function overloading is an example of ——

(i) Inheritance (ii) Run-time polymorphism.

(iii) Compile time polymorphism (iv) All of these.

Ans. : (iii) Compile time polymorphism

29. The range of signed short int is ——

(i) - 128 to 127 (ii) - 32768 to 32767 (iii) 0 to 255 (iv) 0 to 65535

Ans. : (ii) - 32768 to 32767

30. If the value of a = 4 and b = 7, then the value of p after execution of the statement p = - - b + a + + is —— **(March 2002)**

(i) 10 (ii) 11 (iii) 9 (iv) 12

Ans. : (i) 10

31. To read data from a file, the file should be opened in —— mode. **(March 2003, Oct. 2006)**

(i) input (ii) output (iii) append (iv) none

Ans. : (i) input

32. The ability to take more than one form is called ——— in object-oriented programming. (Oct. 2003; March 2011, July 2017)
(i) inheritance (ii) encapsulation (iii) polymorphism (iv) data abstraction

Ans. : (iii) polymorphism

33. Which of the following is not a feature of object-oriented programming ? (March 2004)
(i) Follows bottom-up approach in program design.
(ii) Objects may communicate with each other through functions.
(iii) Follows top-down approach in program design.
(iv) Programs are divided into what are known as objects.

Ans. : (iii) Follows top-down approach in program design.

34. Programming in C++ using classes is called ———programming. (Oct. 2004)
(i) procedure oriented (ii) event drivern
(iii) object oriented (iv) database

Ans. : (iii) object oriented

35. ——— is not a derived data type in C++. (March 2005; Oct. 2010)
(i) Class (ii) Array (iii) Function (iv) Pointer

Ans. : (i) Class

36. ——— is not the feature of OOPs. (Oct. 2005, 09)
(i) Polymorphism (ii) Inheritance
(iii) Data Abstraction (iv) Top-down Approach

Ans. : (iv) Top-down Approach

37. A derived class with several base classes is ——— inheriance. (March 2006, 2009)
(i) single (ii) multiple (iii) multilevel (iv) hierarchical

Ans. : (ii) multiple

38. Which of the following allows to access the private data of other class is ____
(i) In -line function (ii) Friend function
(iii) Main function (iv) All of the above (March 2007)

Ans. : **(ii) Friend function**

39. While accessing the number in a float array using pointer, the pointers value every time increases by _______.
i) 2 ii) 4 iii) 8 iv) 16 (Oct. 2007)

Ans : **(ii) 4**

40. Following operator. cannot be overloaded. (March 2008
i) ++ ii) :: iii) – iv) *

Ans. : **(ii) ::**

41. ——— type of member function of class never takes any argument. (Oct. 2008
(i) Constructor (ii) Destructor
(iii) Operator (iv) None of these

Ans : **(ii) Destructor**

42. For a = 23 and b = 3, the value of c after execution of the statement c = (a/b) * (a%b) will be ______. **(March 2010)**

(i) 14 (ii) 49 (iii) 21 (iv) 69

Ans.: **(i) 14**

43. Out of the following C++ operators, ______ operators can be overloaded. **(March 2012)**

(i) Sizeof (ii) :: (iii) ?: (iv) *

Ans.: **(iv) ***

44. A pointer is a. variable that holds ______ of another variable. **(Oct. 2012)**

(i) Value (ii) Memory Address

(iii Data-type (iv) None of these

Ans.: **(ii) Memory Address**

45. Object Oriented Programming follows ______ approach in program design. **(March 2013)**

(i) top-down (ii) Non-hieranchical (iii) random (iv) Bottom-up

Ans.: **(iv) Bottom-up**

46. When a base class is privately inherited by a derived class, public member of the class becomes ______ of the derived class. **(Oct. 2013; July 2016)**

(i) Public Member (ii) Protected Member

(iii) Private Member (iv) Non-Member

Ans.: **(iii) Private Member**

47. In C++ ______ is an extraction operator. **(Oct. 2014)**

(i) << (ii) >> (iii) && (iv) !

Ans.: **(ii) >>**

48. Object oriented programming uses______ approach of programming. **(March 2015)**

(i) Linear (ii) Non-linear (iii) Top down (iv) Bottom up

Ans.: **(iv) Bottom up**

49. We can not define more than one ______ in a class. **(Oct. 2015)**

(i) Constructor (ii) Destructor (iii) Member Function (iv) Data Member

Ans.: **(ii) Destructor**

50 In C++, double type data consumes ______ bytes in memory.. **(March 2016)**

(i) 2 (ii) 4 (iii) 6 (iv) 8

Ans.: **(iv) 8**

51. ______ symbol is used to declare destructor function in C++. **(July 2018)**

(i) @ (ii) # (iii) ~ (iv) !

Ans.: **(iii) ~**

52. What will be the value of x after execution of following expression in C++?
X = + + m + n + + ; where m = 10 and n = 15. **(March 2019)**

(i) 25 (ii) 27 (iii) 26 (iv) 28

Ans.: **(iii) 26**

53. ________ operator can be overloaded.

(i) + (Plus) (ii) | | (Logical OR)

(iii) % (Modulus) (iv) All i, ii and iii

Ans. : (iv) All i, ii and iii

54. ________ operator cannot be overloaded.

(i) ++ (ii) +

(iii) : : (iv) >>

Ans. : (iii) : :

❑❑

Chapter 4

HTML

Probable marks : 14

Scope of the Syllabus

- Introduction to HTML.
- Why HTML ? Its advantages and drawbacks.
- Study of tags : <HTML>, <HEAD>, <TITLE>,
  <BODY>, <P>, <BR>, <UL>, <OL>, <PRE>, <MARQUEE>
- Font styles
  <B>, <I>, <U>, <BIG>, <SMALL>, <SUB>, <SUP>, <FONT>.
- Image
  HREF, HR, <IMG>, SRC, ALT, HEIGHT, WIDTH, ALIGN
- Tables
  <TABLE>, <CAPTIONS>, <TR>, <TH>, <TD>
- Use of scripting as a language support.
  Note : Only VBscript using for....next, If ... then, MsgBox, InBox, DIM, SET.

INTRODUCTION TO HTML

Q.1 What is HTML ? (Oct. 2002,03 ; Mar. 2006,09,14)

Ans. :

1) It is the most simple, text oriented programming language.

2) HTML stands for **HyperText Mark-up Language**, used to create world wide web document.

3) Using this language user can create web pages which can be viewed in any web browser such as Netscape Navigator or Internet Explorer.

4) **Hypertext** is ordinary text with extra features such as formatting, images, multimedia and links to other documents.

5) **Mark-up** is the process of taking ordinary text and adding extra symbols, such as editor's proof reading symbols. Each of the symbol used for mark-up in HTML is command that tells the browser how to display the text.

6) Mark-up languages are special type of computer languages. They are concerned with only parts of documents according to their functions.

7) They indicate which part of document is title, which is subheading, which is author's name and so on.

8) HTML is essentially a set of instructions to web browser for formatting and layout of web pages.

9) HTML is not a programming language in real sense.

(March 200

Q. 2 Explain features of HTML.

Ans. :

(1) Hypertext Markup Language or HTML is a set of codes that is used to create documen and then it can be published on the World Wide Web.

(2) HTML lets user jump from topic rather than finding and reading information linearly.

(3) Documents prepared in HTML include reference graphics and formatting tags.

(4) HTML is **hyperlink** specification language.

(5) HTML supports to **frames** including target windows and borderless frames.

(6) It contains powerful formatting facilities for text, page, images etc.

(7) It defines the syntax and placement of special embedded directions which are not displayed by the browser but it tells the browser how to display the contents of the document.

(8) It supports for .BMP and animated .GIF images.

(9) HTML support **forms** which make it possible to create documents that collect and process uses input.

(10) It tells how to make a document interactive through special hypertext links.

WHY HTML ? ADVANTAGES AND DISADVANTAGES

Q. 3 What are advantages of HTML ? **(Oct. 2002,03,09, 14 ; March 06,09,14, 19, July 2017)**

Ans. : The advantages of HTML are as enlisted below :

1) For creating HTML document, only text editor is needed. No special software is needed.

2) HTML document can be created on any hardware platform using any text editor.

3) HTML is easy to learn, use of implement.

4) Contains powerful formatting facilities.

5) Required HTML pages can be updated easily, without changing whole document.

6) Any HTML document can be traversed due to hyperlinking facility is available.

7) Independent work can be done and need not to worry about editing programs.

8) If something is not working, then finding error is easy in HTML.

9) HTML will not cost anything for its use. There are no expensive licenses to buy or no upgrades to purchase.

10) Learning HTML is simple than any programming language.

Q. 4 Give the disadvantages of HTML. **(Oct. 2002,03,14 ; March 2006,09, 19, July 2017)**

Ans. :

1) HTML is not a programming language in true sense.

2) Any simple calculation cannot be done in HTML.

3) It cannot be used to display even date.

4) The interactive web pages cannot be built by HTML.

5) The web pages developed in HTML cannot behave like an application.

;) The web pages developed in HTML do not have their own interface.

) Hyperlink is provided in HTML. But for that we need a trip to server at each step.

Q.5 What is HTML ? Name any two softwares that are used for writing HTML codes. State any one advantage and one disadvantage of HTML. **(Oct. 2002)**

Ans. : Please refer Q. No. 1, 3 and 4.

Softwares that are used for writing HTML codes are :

(1) Notepad in Windows

(2) Simple Text in Macintosh

(3) Pico in Unix

Q.6 What are tags ? Explain.

Ans. :

1) A tag is a single unit of mark-up. It is a set of symbols defined in HTML to have special meaning. Tags are instructions that are written directly into text edition.

2) Tags start with a less than sign (<) followed by a keyword and end with greater than (>) sign. These symbols together known as **angle brackets**.

3) The tag part is a **code** usually one or **two letter**, that specify the type of effect.

4) There are two types of tags in HTML :

(i) Start tags (ii) End tags

Start tags are used to begin an effect, and end tags are used to end that effect. Name of end tag is same as that of start tag but the name of end tag is preceded by a forward slash (/).

5) For e.g. <I> is Italic tag. The text written between start tag (<I>) and end tag </I> will be displayed in italic.

<I> HTML </I>

Here, the word 'HTML' will be displayed in italics.

6) Tags can be nested within each other.

For eg. <B><I> Hello </I></B>

7) The attributes are inserted right within the tag that are used to affect tag's behaviour.

Q.7 Give the structure of HTML web page. **(Oct.2011,14, March 2017, 2020)**

Ans. :

1) Every HTML document has the same general structure, and it consists of few tags that define the page as a whole.

2) The primary part of an HTML document are denoted by <HTML>, <HEAD> and <BODY> tags. Each of these tags are known as **Document Structure Tags.**

3) HTML file always starts with <HTML> tag. Similarly ended with </HTML> tag. It declares text within web page viewed in a web browser.

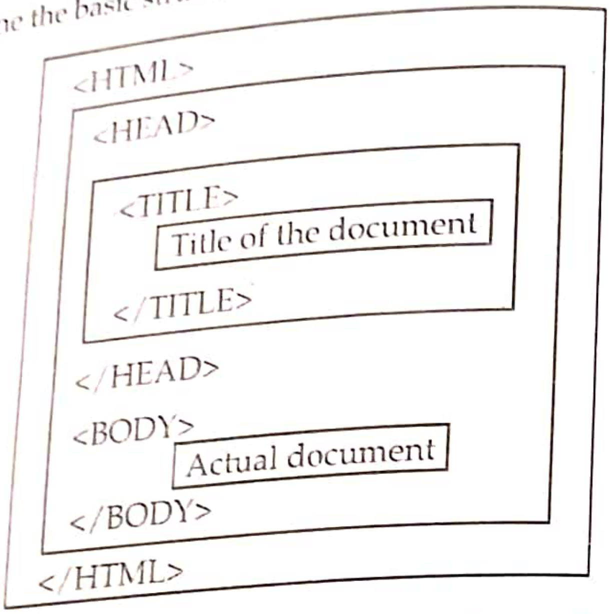
4) HTML document can be divided into two sections :

(a) **The head :** It is like an introduction to the page. It generally consists of title of the page. To define head, add <HEAD> tag at beginning and </HEAD> tag at end of heading.

(b) **The body :** In this user enters the text images and other tags that will actually appear on the web page.

To define the body, place <BODY> tag at beginning and </BODY> tag at the end after the head section.

These tags define the basic structure of every web page.



```
<HTML>
  <HEAD>
    <TITLE>
      Title of the document
    </TITLE>
  </HEAD>
  <BODY>
      Actual document
  </BODY>
</HTML>
```

Q. 8 What are essential and basic tags of HTML code ? OR Explain the purpose of following tags in HTML. (i) <HTML> (ii) <HEAD> (iii) <TITLE> (iv) <BODY>

Ans. :

The essential and basic tags of HTML code are <HTML>, <HEAD>, <TITLE> and <BODY>.

(i) <HTML> :

1) In order for the browser to open the HTML file, it must be told that the file is an HTML file.

2) This can be done by making the beginning of the file with <HTML> tag and end of file with </HTML> tag.

3) All other tags must reside within the <HTML> ..... </HTML> tag.

4) The following example begins and ends a document with HTML tag.

```
<HTML>
        <BODY>
                This is HTML file.
        </BODY>
</HTML>
```

(ii) <HEAD> :

1) <HEAD> tag defines the header area of the page which is not displayed within the page itself in the browser.

2) In the <HEAD> section, <TITLE> tag can be used for give the title for the web page. The end tag </HEAD> ends the header area of the page.

3) For example :

```
<HTML>
        <HEAD>
<TITLE> WELCOME TO FIRST WEB SITE</TITLE>
        </HEAD>
<HTML>
```

(iii) <TITLE> : **(Oct. 2002)**

1) The text between start tag <TITLE> and end tag </TITLE> is the title of the web page and is displayed in the title bar of browser.

2) The title should be descriptive as it is frequently used by web indexing and searching programs to name your web page.

3) There should not be extra space between title tag and text of title.

(iv) **<Body> :** **(March 2008 ; Oct. 2005; July 2019)**

1) The actual contents of the web page that will be displayed on browser will appear in body section of document.

2) The body section starts with <BODY> tag and ends with </BODY> tag.

3) There are several optional attributes for this tag, such as set background images, change the font of text by using attributes.

4) The <BODY> .... </BODY> section defines the actual instructions for laying out graphics, text, multimedia and other elements in the browser's work's area.

Q. 9 What are different attributes of body tag in HTML?

Ans.: The actual contents of the web page that will be displayed on browser will appear in body section of document. The body section starts with <body> and end with </body>. The body tag can have a number of attributes which allow you to specify a background image or background and foreground colour.

1. **Background attribute:**

   This attribute specify an image file which will be used as the background. It is important to make an image file as small as possible, because if you use a large file as a background your page will appear to very slow to download.

   Ex. <BODY BACKGROUND ="C: \ Ganesh.jpg">Ganesh Pooja</BODY>

   Here image Ganesh.gif will be set as background to our web page.

2. **Background colour:**

   BGCOLOR attribute specifies the colour to use for the background.

   Ex. BGCOLOR = "#rrggbb" sets the background colour to RGB value where RR, GG & BB are hexadecimal colour codes for Red, Green and Blue levels from 0 to 255 or 00 to FF. The colour "000000" is black while "FFFFFF" is white. We can also place name of colour in double quote as BGCOLOR = "Blue"

3. **Text attribute:**

   This attribute defines the colour want for text.

   Ex. <BODY TEXT="RED">WELCOME TO TEA PARTY</BODY> word "WELCOME TO TEA PARTY" displays in RED colour.

4. **LINK attribute:**

It sets colour for hyperlinks. This attribute defines colour in which to display Hyper Te links which have not been visited.

Ex. <BODY LINK="00FF00">LINK TO Page 2</BODY>

Sets the default text colour of hypertext anchors to green colour.

5. **VLINK attribute:**

This defines colour for links which have previously been visited.

Ex. <BODY VLINK = "00FF00">Name</BODY>

Sets the default text colour of visited hypertext link to colour green.

Q. 10 Explain the attributes, BGCOLOR and BACKGROUND of <BODY> tag. **(Oct. 2008,**

Ans. :

1) **BGCOLOR**

i) The BGCOLOR attribute of the <BODY> tag changes the background colour of th web page and sets it to the colour specified within tag value.

ii) For eg. : <BODY BGCOLOR = "RED">

The above tag sets background colour of webpage to Red.

2) **BACKGROUND**

i) The BACKGROUND attribute of the <BODY> tag is used to put an image c wallpaper in the background of the web page.

ii) For eg. : <BODY BACKGROUND ="imagename">

Q. 11 Explain the procedure to prepare and view HTML document.

Ans. : The procedure to prepare and view HTML document is as follows :

Step 1 : Open a text editor e.g. : Notepad.

Step 2 : Write the appropriate HTML code.

Step 3 : Save the HTML code in a file having extension HTML.

Step 4 : Open the Browser e.g.: Internet Explorer.

Step 5 : Browse the file or type in the appropriate address of the HTML file in the address ba to view the HTML page.

Q. 12 Explain following tags in HTML.

(1) <H1> .... <H6> tag (2) <ADDRESS> tag

Ans.: (1) <H1> ...<H6> tag:

Heading tags are used to define the levels of the header. There are six heading in HTM these are HI., H2, H3, H4, H5 and H6. <HI> heading is displayed in larger, bolder font tha low level heading i.e. <H2> ... <H6>.

Ex. <H1>Notes of computer Science</H1>

Output displayed will be: **Notes of Computer Science**

Heading tag can be used following attributes: ALIGN = "LEFT" ALIGN = "RIGHT", ALIGN="CENTER", ALIGN="JUSTIFY"

Ex. <H6 ALIGN="CENTER">Kolhapur</H6>

It's webpage display word "Kolhapur" at centre of page.

(2) **<ADDRESS> tag:** Text contained in an ADDRESS tag should contain information used on ADDRESS tag to put in your name, email address, home page location, street address. ADDRESS tag is displayed in italic text.

```
Ex. <ADDRESS><B>X.Y.Z.</B>
        <BR><U>Bombay</U>
        <BR><B>(020) 41112<B></ADDRESS>
```

Output will be - ***X.Y.Z.***

<u>Bombay</u>

(020) 411112

The text in address tag is recognised by search engine as your address information. The address tag is usually displayed in italic text.

Q. 13 Explain the following tags in HTML : **(March 2007,11 ; Oct. 2005,06)**

(i) <P> tag (ii) <BR> tag (iii) <HR> tag (iv) <PRE>

Ans. : (i) <P> tag :

1) <P> tag is used for creating paragraph of text.

2) A paragraph can be created by enclosing text within paragraph codes <P> and </P>.

3) The browser ignores the paragraph created by user while writing codes by pressing 'Enter'. User must specifically define a paragraph in the code by using a paragraph tag.

4) The <P> tag has one optional attribute called align. It is used to specify where the text appear on screen.

   (i) <P align = left> Left alignment like normal text.

   (ii) <P align = right> : Text is aligned to right margin, but not justified to left.

   (iii) <P align = center> : Text is centered.

   (iv) <P align = Justify> : Text is justified to left and right margin.

(ii) <BR> tag : **(March 2012, 16 ; Oct. 2002, 2004)**

1) <BR> tag insert line break into a text flow.

2) It tells the browser to wrap the text that follows onto a new line without inserting any extra space between the lines.

3) e.g. Sonia Gandhi, <BR> 10 Janpath, <BR> New Delhi.

In above example, the matter will be displayed as :

Sonia Gandhi,
10 Janpath,
New Delhi

4) For an entire blank line <BR> tag on that line. For multiple blank lines, just count them and type <BR> tag on every expected blank line.

(iii) <HR> tag : **(Mar. 2016; Oct. 2006, July 2016)**

Please refer Q. 15(c), Ch. 4, Pg. 4-9.

(iv) <PRE> tag : **(March 2004,07,12 ; Oct. 2003,06,14; July 18)**

(1) <PRE> is used to preformat the text. The text appearing between start tag <PRE> and end tag </PRE> is displayed in mono space form.

(2) This tag is used to position the characters.

(3) This tag displays the text in exactly same format as the character and line spacing format defined in source HTML document.

(4) For example :

```
<PRE>
                      1
                 1         1
            1         2         1
       1         3         3         1
    </PRE>
```

The display will be as :

```
                      1
                 1         1
            1         2         1
       1         3         3         1
```

Q. 14 How text is formatted by using <B>, <I> and <U> tags ?

Ans. : To give additional emphasis to the text, HTML provides bold face, italics and underlining the words. This can be done by using <B>, <I> and <U> tags as :

(i) <B> tag : **(Mar. 2005)**

<B> is bold tag. The text appearing between start tag (<B>) and end tag (</B>) will be displayed in bold letters.

e.g. <B> Bold Text </B> ⇒ **Output : Bold Text**

(ii) <I> tag : **(Mar. 2013)**

<I> is italic tag. The text appearing between start tag (<I>) and end tag (</I>) will be displayed in italics.

e.g. <I> Italic Text </I> ⇒ **Output :** *Italic Text*

(iii) <U> tag : **(Oct. 2007, July 2016)**

<U> is underline tag. The text appearing between <U> and </U> tag will be underlined. Generally, browser indicates hyperlinks in web pages by underlining them. So it is generally avoided.

e.g. <U> Underlined Text </U>. ⇒ **Output :** <u>Underlined Text</u>

Q. 15 Explain the purpose of following tags in HTML with example.
(a) <marquee> (b) <LI> (c) <HR> **(March 2002,04,08 ; July 2019)** **(Oct. 2002,14)**

Ans. : (a) <marquee> :

(1) It is used for scrolling the text and images on screen from right to left.

(2) The text written between start tag <marquee> and end tag </marquee> will scroll on screen, in horizontal line.

For e.g. : <marquee> computer science </marquee>

Here, the word computer science will scroll on screen.

(3) There are several attributes associated with <marquee> such as BGCOLOR, HEIGHT and WIDTH.

(4) This is used to scroll the current news or position of different companies in stock market.

(b) **<LI>**

A tag <LI> is used to indicate actual list elements.. It is used inside <UL> and <OL> tag to show unordered and ordered list element. **(March 2013)**

Example :

Input

```
<UL>
<LI> Primary
<LI> Secondary
<LI> Higher secondary
</UL>
```

Output

- Primary
- Secondary
- Higher secondary

(c) **<HR> :** **(March 16; Oct. 06, July 2016)**

(1) <HR> tag is horizontal rule tag, also called as horizontal line.

(2) A web page can be divided into separate sections by using horizontal rule <HR> tag

(3) This tag is mostly used for decorative purposes.

(4) <HR> tag takes several attributes such as COLOR, SIZE, WIDTH align etc.
e.g. : <HR SIZE = "6" WIDHT = "60%" align = "right">
This displays a line of six pixels thick that spans 60 percent of the browser's window and right aligned.

(5) The default align is to center in the window.
ex-< HR NO SHADE > This means for an unshaded horizontal line.

Q.16 What is <EM> tag ? What is the difference between <EM> tag and <I> tag ? **(March 2015, July 2018)**

Ans. :

1) <EM> is emphasis tag. The text appearing between start tag (<EM>) and end tag (</EM>) will be displayed in italics. This tag is used to empasize the text.

2) The main difference between <EM> tag and <I> tag is that text to speech browsers gives spoken emphasis to the text within emphasis tags, while no such emphasis is given to text within italic tags.
For e.g. <P> you <EM> must </EM> handover the money to him </P>.

Q.17 What is <STRONG> tag? What is the difference between <STRONG> and <B> tags?

Ans. :

1) <STRONG> is strong tag. The text appearing between start tag (<STRONG>) and end tag (</STRONG>) will be displayed in bold.

2) This tag is used to create strong emphasis.

3) The main difference between <STRONG> and <B> tags is that the text to speech readers gives strident pronunciation to the strong text, while no such strident pronunciation is given to bold text.
For eg. <P> If they don't give me that raise <STRONG> tomorrow </STRONG>, I quit </P>.

Q.18 What are types of List? Explain with example?
OR
Explain <OL> and <UL> tag used in HTML with example. **(March 2018)**

Ans.: There are three main types of list - unordered list, ordered list and definition list.
Different list types are :
<OL> Ordered list

<UL> unordered list

<DL> Definition list

1. **Unordered List** : The simplest type of list is an unordered list. The elements of unordered list are displayed as a series of bullet points. These list is contained between <UL> and </UL> tags. Each element in the list should be started by an <LI> tag. Unordered list have several bullet styles like a closed circle [●], open circle [○] or square [■].

```
Ex. : <UL>
          <LI>Element 1</LI>
          <LI>Element 2</LI>
          <UL>
                <LI>Element 3.1</LI>
                <LI>Element 3.2</LI>
          </UL>
     </UL>
```

So output given by browser is;

```
•    Element 1
•    Element 2
     ○    Element 3.1
     ○    Element 3.2
```

Attribute of unordered list are <UL type ="circle"> <UL type = "disc"> and <UL type ="square">

2. **Ordered List:** An ordered list is .also a list of items. The list items are marked with numbers'. Ordered list start with <OL> and close with </OL>. Attribute use with ordered list are:

Type = 1 (Arabic numbers)
Type = I (Upper case roman numbers)
Type = a (lowercase alphanumeric)
Type = A (Uppercase alphanumeric)
Type = i (lowercase Roman numbers)
START = X

Ex. <OL START="5">would start your ordered list numbering with the number five.

```
Example : <OL>
               <LI>milk
               <LI>bread
               <LI>cheese
          </OL>
```

Output given by browser is,

```
1.   milk
2.   bread
3.   cheese
```

3. **Definition list:** It uses <DL> start tag and </DL> end tag to create definition list. <DD> tag is used for definition themselves and <DT> is used for definition terms. Definition list is not a list of items but list of terms and explanation of the terms. **(March 2015)**

```
Ex. <DL>
        <DT>Computer</DT>
        <DD>IBM</DD>
        <DT>T.V. </DT>
        <DD>Videocon</DD>

    </DL>
```

Output on browser is,

```
Computer
    IBM

T.V.
    Videocon
```

Q. 19 How can unordered lists be created ? **(Oct. 2006, Mar. 2007)**

Ans. :

1) An unordered list is a list of items that have no particular order or sequence.

2) Unordered lists requires start and end tags (<UL> and </UL>).

3) A special tag <LI> is used to indicate actual list elements.

4) Unordered list are bulleted lists. They can be preceded by one of the several bullet styles like a closed circle (•), or an open circle (O) or a square(■).

e.g.

```
<UL>
<LI> Eggs
<LI> Milk
<LI> Apples
</UL>
```

The list will be displayed as,

- Eggs
- Milk
- Apples

5) Both <UL> and <LI> has same set of attributes given below.

TYPE = "CIRCLE"
TYPE = "DISC"
TYPE = "SQUARE"

The CIRCLE attribute value is used for hollow circle (O), the DISC type creates a solid bullet (•) while SQUARE value creates a solid block (■). The default appearance for a list is with disc.

6) The end tag (</UL>) is always required at the end of unordered list. Also use </LI> at the end of each list item.

Q. 20 How ordered lists are created ? OR Explain the use of <OL> tags with example. **(March 2005, Oct.2014)**

Ans. :

1) An ordered list is used when the sequence of the list of items is important.

2) Ordered lists are numbered in some fashion. Ordered lists can be preceded by Arabic numerals, upper case or lower case Roman numerals, or upper case or lower case alphanumeric characters.

3) The tags for an ordered list are <OL> (starting tag) and </OL> (ending tag). Also <LI> tag is used to indicate actual list element.

4) e.g.

```
<TITLE>
        Shopping List
</TITLE>
<BODY>
<OL>
        <LI> Eggs
        <LI> Milk
        <LI> Apples
        </OL>
</BODY>
```

The list will be displayed as :

```
1.  Eggs
2.  Milk
3.  Apples
```

5) The attributes that can be used with ordered lists are :
TYPE = "1" (Arabic numbers)
TYPE = "a" (lowercase alphanumeric)
TYPE = "A" (uppercase alphanumeric)
TYPE = "i" (lowercase Roman numbers)
TYPE = "I" (uppercase Roman numbers)

6) The default appearance for list is with arabic numbers.

7) There is another attribute START, which allows us to establish the beginning of list number sequence. It has the form -
START = "number"

Q. 21 Explain the use of following tags in HTML. **(Oct. 201**

Ans. : 1) <A> tag :

1) <A> is 'anchor' tag. It is used to create links or hyperlinks.

2) Links point to different files on the web.

3) The text or image enclosed between starting tag (<A>) and ending tag (</A>) is a link

4) This link is clickable in a graphical browser.

5) With most browsers, the text within these tags is displayed in a different colour and underlined.

e.g. <A HREF = "http:\\www.rosecards.com">
More Cards </A>

6) The object to which the link has to be made is defined by the HREF attribute. HREF refers to hypertext reference. It defines URL (Uniform Resource Locator) of the destination of anchor.

2) **<SUB> tag :** **(March 2004, 08, 12 ; Oct. 2004, 05, 07 ; July 2019)**

1) <SUB> is subscript tag.

2) The text enclosed within _{and} is displayed in subscript form.

3) It is bit lower than text.

4) This is useful for chemical formulae.

e.g. H₂ 0

It will be displayed as : H_2O

3) **<SUP> tag :** **(March 2004, 2005; Oct.2010; July 2018)**

1) <SUP> is superscript tag.

2) The text enclosed within start tag (^{) and end tag (}) will be displayed in superscript form.

3) It is a bit higher than the normal text.

4) It is useful for mathematical formulae.

e.g. E = mc²

It will be displayed as :

$E = mc^2$

4) **<Font> tag :** **(Oct. 2015)**

1) <FONT> is font tag.

2) It is used to format the size, type face and colour of enclosed text.

3) The <FONT> tag can be used with three different attributes : SIZE, FACE and COLOR.

4) The SIZE attribute can be specified in absolute or relative values ranging from 1 to 7. Using a relative font size i.e. by putting plus or minus sign before the number will change the font size relative to the default size.

5) The COLOR attribute is specified with a RGB code or specify a color name.

6) The FACE attribute specifies a type face that is used for the text enclosed by the font element.

5) **<BIG> tag :** **(Oct.2010)**

1) <BIG> is big tag.

2) The text enclosed within starting tag (<BIG>) and ending tag (</BIG>) is displayed in larger font.

3) <BIG> tag has the same effect as <FONT SIZE = "+1">.

4) If already the size is largest, tag is ignored.

(Oct.2010)

6) **<SMALL> tag :**

1) <SMALL> is small tag.

2) The text enclosed in <SMALL> (starting tag) and </SMALL> (ending tag) is displayed in smaller font.

3) <SMALL> tag has same effect as <FONT SIZE = "– 1">.

4) If already the size is smallest, then the tag is ignored.

7) **<STRIKE> tag :**

1) <STRIKE> is strike tag.

2) The text enclosed within the tags <STRIKE> and </STRIKE> or <S> and </S> would have a line drawn through the middle of the text.

e.g. <S> You are mad </S>

It will be displayed as :

8) **HREF :**

1) The HREF attribute is used with <A> i.e. anchor tag.

2) HREF refers Hypertext Reference.

3) This attribute marks the anchor as the start of a link to another document or resource or to a particular place in another document.

4) For e.g.

<A HREF = "http://www.rediff.com">

Enter your Email-id </A>

In the above e.g. "Enter your Email-id" is the hypertext link to the website indicated by URL specified i.e. rediff.com.

Q. 22 Explain anchor tag with example. OR

How can you have a link to an image by having image as a link? OR

Explain hyperlink with example? OR

List any three types of hyperlinks with suitable example. **(March 2003)**

Ans.: Hypertext links done using <A> or anchor tag.

<A> tag looks as <A HREF="NOTE">. The Anchor</A> where HREF is hypertext reference attribute defines URL (uniform resource locator) of the destination of anchor. The text between <A> and </A> is displayed by browser in Underline. Thus anchor is a piece of text or some other object which marks the beginning or end of hypertext link. HREF attribute marks the anchor as the start of a link to another document or resource to a particular place in another document.

1. **Link to a page on the world wide web:**

<A HREF=http://www.hotmail.com.>send email</A>

Here "send email" is a hypertext displayed in web browser which link to website hotmail.com after clicking on hypertext.

2. **Link to a image by image as link:**

<A HREF="India.jpg"><IMG SRC="indiasmall.jpg"></A>,

Here image indiasmall.jpg is a hypertext link to the image"file located in same directory

These image file is displaced as a button or small icon in web browser. After clicking on this image large version i.e. india.jpg file will be open.

3. **Link to document located in different directory:**

<A HREF="C: \ program \ data.html>click here to see another file</A>

Here by clicking on "click here" word destination page will be displayed which specified in the path.

4. **Link to the same page:**

<A Name = "step 2"><H2>Same page hyperlink</H2></A>

Selection area in an HTML can be marked as destination of hypertext links via a NAME attribute.

<A HREF="#step2">link to same page</A>

Here link can be created to display web page on web browser. This type of anchor called named anchor because to create the links, you insert HTML names within tilt document.

Q.23 Write a short note on RGB codes.

Ans.:

1) Colors in HTML can be specified by a color name or by color code, known as RGB code.
2) In RGB codes, R stands for Red, G stands for Green and B stands for Blue.
3) There are 16 color names is a HTML. They are BLACK, SILVER, GRAY, WHITE, MAROON, RED, PURPLE, FUCHSIA, GREEN, LIME, OLIVE, YELLOW, NAVY BLUE, TEAL and AQUA.
4) RGB codes are always 6 numbers. First two numbers specify amount of red. Next two numbers specify amount of green and last two numbers specify amount of blue.
5) By mixing these three primary colors in different amount, it is possible to derive any color.
6) RGB codes use hexadecimal numbering system.
7) For Red color, RGB code is # FF0000, for Blue color RGB code is # 0000FF and for green color, RGB code is # 00FF00.
8) Yellow is the combination of maximum red and maximum green. Hence, RGB code of yellow is # FFFF00.
9) Similarly, RGB code for Black color is # 000000 and RGB code for White is # FFFFFF.

Q.24 What is <IMG> tag ? What are the attributes that can be used with <IMG> tag ?

(March 15, 16 ; Oct. 2002)

Ans. :

1) <IMG> is image tag. Its purpose is to include graphic images in the body of the web page.
2) There are two types of images :
   i) Inline images : It occur in the middle of a line of text. If a image is large one, then the line becomes very tall.
   ii) Floating images : It cause text to wrap around the image. The paragraph will flow around the image for several lines, if the image is large.

3) To make an image as a separate paragraphs, it is enclosed within paragraph elements.
e.g. <P> <IMG SRC = "C : \My Documents \ Lion.jpg"> </P>

4) Generally SRC and ALT attributes are always used with <IMG> tag. SRC attribute includes appropriate path of image file for searching. For any browser, that is not displaying images, the alternate text contained inside the ALT attribute is displayed instead of the image.
e.g. <IMG SRC = "Mickey mouse.jpg" ALT = "MICKEY">

5) In addition, alignment attributes and sizing attributes are used with <IMG> tag as :

a) For Inline images, alignment attribute has three attribute values, that are ALIGN = "TOP", ALIGN = "MIDDLE" and ALIGN = "BOTTOM".

b) For Floating images there are two attribute values, which are ALIGN = "LEFT" and ALIGN = "RIGHT".

6) To indicate exact size of image use WIDTH and HEIGHT attributes.
e.g. <IMG SRC = "Preyas.jpg" WIDTH = "25" HEIGHT = "120">

7) Display image with border using BORDER attribute.
e.g. BORDER = "1" then thin border will appear around the image.

Q. 25 What is <TABLE> tag ? What are the attributes that can be used with <TABLE> tag ?

Ans. :

1) A table can be created using <Table> .....</Table>.

2) A table consists of rows and columns (also called as cells.)

3) The row is to be defined first and then cells or columns are inserted into the row from left to right.

4) The row is created using <TR>....</TR> tag while the cells can be inserted in the row using <TH>....</TH> or <TD>....</TD> tags.

5) Normally <TH> tag is used just to give the heading or title to the column or row or both.

6) <TD> is used for displaying actual data. Structure of a simple table is given below :

Name of the Attribute	Description	Example
1. Border	Draw an outline around table row and cell. By default table has no border.	<Table border = "2">
2. Width	Defines how wide your table will appear across the width of the screen	<table width = "50"> absolute value in pixels <table width = "80%"> percentage value of document width.
3. Cellpadding	Decides the amount of spacing between each cell border and the actual content of cell.	<table cellpadding = "2" border = "2">
4. Cellspacing	Creates space between the cells.	<table border = "4" cellspacing = "4">

Name of the Attribute	Description	Example
5. Bgcolor	To give the background color to the table	<table border = "4" bgcolor = "red">

6. Bordercolor	Give the color to the border	<table border = "4"
		bgcolor = "red"
		bordercolor = "black">

7. Align	Decides the alignment of the table. Default is left.	<table align = "center">

Attributes of <TR> <TD> <TH> tags as follows :

Attribute	Description	Example
1. Align	Specifies the horizontal alignment of row or column. Default alignment is left.	<tr align = "center"> <td align = "right">
2. Valign	Specifies the vertical alignment of row or column. Default alignment is left.	<tr valign = "center"> <td valign = "right">
3. Bgcolor	Specifies background color of specific row or column.	<tr bgcolor = "aqua"> <td bgcolor = "brown">

Attributes of table cell only as follows :

1. colspan	Number of columns that a cell spans.	<td colspan = "2">
2. rowspan	Number of rows that a cell spans.	<td rowspan = "2">

Q.26 Explain the use of following tags in HTML.
1. <CAPTION> tag 2. <TR> tag 3. <TH> tag 4. <TD> tag

(March 2005)

Ans.: **1) <CAPTION> tag :**

1) <CAPTION> tag is used to create a caption on top of or below the table.
2) It requires <CAPTION> start tag and </CAPTION> end tag.
3) It can be used with ALIGN attribute by setting it to TOP or BOTTOM.
4) Caption is displayed outside the table's border.

e.g.
```
<TABLE>
<CAPTION ALIGN = "TOP">
Yesterday's Weather
</CAPTION>
```

2) **<TR> tag :**

1) <TR> is table row tag. The start tag is <TR> and end tag is </TR>.
2) It creates a horizontal row of cells and contains table headings or table data.
3) Each use of a table row element (tag) begins a new table row.
4) A row must contain atleast one table data element or table heading element.
5) It includes attributes like ALIGN, BGCOLOR and VALIGN.

3) **<TH> tag :** (Mar. 2013

1) <TH> is table heading tag. The start tag is <TH> and the end tag is </TH>.
2) This tag is used to represent individual column heading of a table.
3) By default text in this cell is bold and centered.
4) It has ALIGN, VALIGN, CELLSPACING, CELLPADDING etc. attribute to set the heading content (LEFT, RIGHT, CENTER)

4) **<TD> tag :** (July 2016

1) <TD> is table data tag. The start tag is <TD> and the end tag is </TD>.
2) <TD> creates each individual cell.
3) The number of cells in a row determines the number of columns.
4) It includes ALIGN, VALIGN, CELLSPACING, CELLPADDING etc. attribute.
5) By default text in this cell is aligned left and centered vertically.

Q. 27 What are COLSPAN and ROWSPAN attributes ?

Ans. :

1) COLSPAN and ROWSPAN are special attributes that can be used with <TH> and <TD> tags.
2) There may be some situation, in which one cell of table to span more than one row or column. For such situations, COLSPAN and ROWSPAN attributes may be used.
3) The COLSPAN attribute can be used to make cell contents merge with another cell.

   e.g. <TD COLSPAN = "2">
   <TH COLSPAN = "2">

   This spans two columns.
4) The ROWSPAN specifies how many rows a cell should take up.

   e.g. <TD ROWSPAN = "2">
   <TH ROWSPAN = "2">

VB SCRIPT

Q. 28 What is script ? Explain VB script. Give one example.

Ans. :

1) A script means a series of commands that will be executed by host environment (or server).
2) Scripting enables us to set and store variables, performs operations on variables.
3) By integrating script, make static HTML page active or dynamic, so that it is called as DHTML i.e. dynamic hyper text markup language.
4) Scripting languages are special programming languages. These are used on web page to control different elements of the page including controls frames and browser interface.
5) There are two famous scripting languages :

   1) VB Script 2) Java Script

6) VB Script allows web author to write small scripts, that will be executed on users browsers rather than on the servers.

e.g. An application collects data from a form and then sends it to the server. If it can validate the data for completeness and correctness before sending it to the server, it will greatly improve the performance of browsing section. Since data is sent to server only if it is verified as correct.

7) Another importance of VB Script is increased functionality introduced to the web authors in the form of applet, plugs-in, Active (x)-control and objects. Each of these things can be used to add extra functions and interactivity to the web page.

e.g.
```
<HTML>
<HEAD>
<TITLE> HELLO WORLD </TITLE>
<SCRIPT LANGUAGE = "VBSCRIPT">
SUB can( )
            Msgbox "HELLO WORLD"
            END SUB
            </SCRIPT>
            </HEAD>
            <BODY>
            <FORM NAME = "FORM 1">
            <INPUT TYPE = "BUTTON" NAME = "B1"
                    VALUE = "CLICK" ONCLICK = "can( )">
            </FORM> </BODY>
            </HTML>
```

Q.29 What is <SCRIPT> tag ?

Ans. :

1) The VBSCRIPT can be placed in <SCRIPT> start tag and </SCRIPT> end tag. It contains attribute LANGUAGE = "VBScript" as shown below :

```
<SCRIPT LANGUAGE = "VBScript">
            (Scripting code)
</SCRIPT>
```

2) When browser hits <SCRIPT> tag, it calls VBScript interpreter to compile and execute the code.

3) The code is placed in event handlers but procedures can also be included.

4) Scripting allows to take control of contents of a page and manipulate them with the program.

Q.30 Give the advantages (features) of VBScript.

Ans. :

(i) **VBScript designed to be fast :**

VB Script does not support any strict data type (i.e. integer, character, float etc.). The only data type available is variant. It is a special all purpose data type, which can be used to store any kind of data.

(ii) VB Script is safe :

It provides file system components which consists of objects, which can be used to perform I/O. But, all language features that enable access to local files on which application is running has been removed from VB script to enhance the security.

(iii) Rich set of functions :

VB Script provides with rich set of functions for performing various types of operations via array, manipulating functions, data and time functions, string functions, mathematical functions, conversion functions.

(iv) Error handling :

VB Script allows user to access and handle errors through the use of Err object.

Q. 31 What are the limitations of VB Script ?

Ans. :

(i) **Array handling :** All VB Script arrays are zero based. It does not allow user to change the base of an array variable for specific implementation.

e.g. If we want to create array of top ten students in a class, it would make sense to start array from 1. But we must start array from zero.

(ii) **VB Script cannot create user defined data type :** VB Script provides with certain objects and collections, but does not give provision for a user to create user defined data type.

(iii) Dynamic data exchange (D.D.E.) : Is not supported in VB Script, as it may violet the integrity of server.

Q. 32 Write HTML code to create a birthday card. The card must have appropriate message and link to www.archiesonline.com

Ans. : HTML code is as follows :

```
<HTML>
    <HEAD>
        <TITLE> BIRTHDAY CARD </TITLE>
    </HEAD>
    <BODY BGCOLOR = "AQUA" TEXT = "MAROON">
    <CENTER>
        <H2><U> ON YOUR BIRTHDAY <BR>
            WITH LOTS OF LOVE <BR></U>
        </H2>
        <FONT FACE = "BOOKMAN OLD STYLE" size = "+5">
        <EM>
                    Smiling Eyes <BR>
                Worm Hearts, Loving Thoughts, <BR>
                    Beautiful Moments <BR>
                    Special Wishes...<BR>
                Are all about one person, one day <BR>
                    One occassion <BR>
                <B> You and your Birthday </B><BR>
                <H2> Have A Good Time ! </H2>
            </EM>
            <STRONG>
                <BR> Wish You a <BR>
                <H1>
                    <A HREF = "http :\www.archisonline.com">
                    HAPPY BIRTHDAY </A>
                </H1>
            </STRONG>
        </CENTER>
    </BODY>
</HTML>
```

Q. 33 Write an appropriate HTML code to display the following.

Ans. :

C++ Data types

1. Built in
    1. Integral
        1. Integer
        2. Char
    2. Floating
        1. Float
        2. Double
    3. Void
2. User defined
    1. Structure
    2. Class
    3. Union
    4. Enumeration
3. Derived
    1. Arrays
    2. Functions
    3. Pointers

HTML code like as follows :

```
<HTML>
<HEAD>
<TITLE>C++ Data types </TITLE>
</HEAD>
<BODY>
<H2 ALIGN = "CENTER">
C++ Data types </H2>
```

```
        <OL TYPE = "1">
        <LI> Built in
            <OL TYPE = "1"> <LI> Integral
        <OL TYPE = "1">
            <LI> Integer
            <LI> Char
            </OL>
        <LI> Floating
            <OL TYPE = "1">
            <LI> Float
            <LI> Double
            </OL>
        <LI> Void
        </OL>
    <LI> User defined
        <OL TYPE = "1">
        <LI> Structure
        <LI> Class
        <LI> Union
        <LI> Enumeration
        </OL>
    <LI> Derived
        <OL TYPE = "1">
            <LI> Arrays
            <LI> Functions
            <LI> Pointers
        </OL>
</OL>
</BODY>
</HTML>
```

Q. 34 Write a HTML code for following.

CRICKET ANALYSIS

COUNTRY	PLAYED	WON	Lose
INDIA	30	23	07
AUS	24	19	05
PAK	18	02	16
ZIM	10	07	03

Each country name should have a link to B.HTML, which contains list of players of appropriate country. Write code for both files.

Ans. : *HTML code for* A.HTML files is as follows :

```
<HTML>
<HEAD>
<TITLE> A.HTML </TITLE>
</HEAD>
<BODY>
<TABLE BORDER = 3>
<CAPTION ALIGN = "TOP">
CRICKET ANALYSIS
</CAPTION>
<TR>
        <TH> COUNTRY </TH>
        <TH> PLAYED </TH>
        <TH> WON </TH>
        <TH> LOSE </TH>
</TR>
<TR>
        <TD> <A HREF = "B.HTML"> INDIA </A> </TD>
        <TD> 30 </TD>
        <TD> 23 </TD>
        <TD> 07 </TD>
</TR>
<TR>
        <TD> <A HREF = "B.HTML"> AUS </A> </TD>
        <TD> 24 </TD>
        <TD> 19 </TD>
        <TD> 05 </TD>
</TR>
<TR>
        <TD> <A HREF = "B.HTML"> PAK </A> </TD>
        <TD> 18 </TD>
        <TD> 02 </TD>
        <TD> 16 </TD>
</TR>
<TR>
        <TD> <A HREF = "B.HTML"> ZIM </A> </TD>
        <TD> 10 </TD>
        <TD> 07 </TD>
        <TD> 03 </
</TR
```

Q. 35 Write a HTML code for a web page displaying the following.

Yeshwant College Nanded	
Course	Capacity
B.Sc. (comp)	80
B.Sc. (C.A.)	80
M.Sc. (Comp)	30
M.C.M	40

Ans. : HTML code is as follows :

```
        <HTML>
            <HEAD>
            <TITLE> COURSES IN COMPUTER SCIENCE
            </TITLE>
        </HEAD>
        <BODY>
        <TABLE BORDER = "1">
        <TR>
            <TH COLSPAN = "2">
                YESHWANT COLLEGE, NANDED
            </TH>
        </TR>
        <TR>
            <TH> Course </TH>
            <TH> Capacity </TH>
        </TR>
        <TR>
            <TD> B.Sc (comp) </TD>
            <TD> 80 </TD>
        </TR>
        <TR>
            <TD> B.Sc (C.A.) </TD>
            <TD> 80 </TD>
        </TR>
        <TR>
            <TD> M.Sc (comp) </TD>
            <TD> 30 </TD>
        </TR>
        <TR>
            <TD> M.C.M. </TD>
            <TD> 40 </TD>
        </TR>
        </TABLE>
        </BODY>
    </HTML>
```

Note : Here, 'Yeshwant College Nanded' is given within the table border. Hence, we cannot use <CAPTION> tag to write it because, the text written within <CAPTION> and </CAPTION> is displayed outside the table border.

Q. 36 Write a HTML code to display the following :

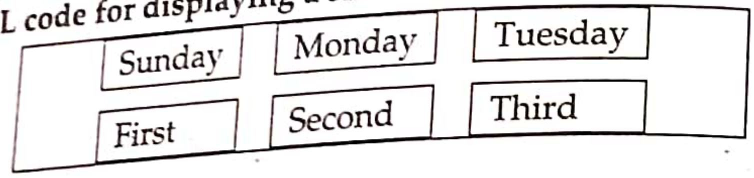
Sr. No.	Student Name	Marks Obtained			Total
		Test 1	Test 2	Test 3	
1	Maheshwari	150	150	150	450
2	Akansha	129	130	131	390
3	Asma	125	115	120	360

Ans. : HTML code is as follows :

```
    <HTML>
    <HEAD>
    <TITLE> Merit list </TITLE>
    </HEAD>
    <BODY>
    <TABLE BORDER = "2">
        <TR>
             <TH ROWSPAN = "2"> Sr. No. </TH>
             <TH ROWSPAN = "2"> Student <BR> Name </TH>
             <TH COLSPAN = "3"> Marks Obtained </TH>
                  <TH ROWSPAN = "2"> Total </TH>
                  </TR>
        <TR>
             <TH> Test 1 </TH>
             <TH> Test 2 </TH>
             <TH> Test 3 </TH>
        </TR>
        <TR>
             <TD> 1 </TD>
             <TD> Maheshwari </TD>
             <TD> 150 </TD>
             <TD> 150 </TD>
             <TD> 150 </TD>
             <TD> 450 </TD>
        </TR>
             <TR>
             <TD> 2 </TD>
             <TD> Akanksha </TD>
             <TD> 129 </TD>
```

```
                    <TD> 130 </TD>
                    <TD> 131 </TD>
                    <TD> 390 </TD>

            </TR>
            <TR>
                    <TD> 3 </TD>
                    <TD> Asma </TD>
                    <TD> 125 </TD>
                    <TD> 115 </TD>
                    <TD> 120 </TD>
                    <TD> 360 </TD>

            </TR>
        </TABLE>
    </BODY>
    </HTML>
```

Q. 37 Write a HTML code for displaying a six-celled table.



Sunday	Monday	Tuesday
First	Second	Third

Ans. :

HTML code is as follows :

```
    <HTML>
    <HEAD>
          <TITLE> Celled Table </TITLE>
    </HEAD>
    <BODY>
    <TABLE BORDER = "3" CELLSPACING = "50">
          <TR>
                <TD> Sunday </TD>
                <TD> Monday </TD>
                <TD> Tuesday </TD>
       </TR>
       <TR> <TD> First </TD>
                <TD> Second </TD>
                <TD> Third </TD>
       </TR>
  </TABLE>
  </BODY>
  </HTML>
```

Q. 38 Write HTML code for the following : **(March 2002)**

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="3">YEAR</td></tr>
<tr><td>1998</td><td>1999</td><td>2000</td></tr>
<tr><td rowspan="2">Sales</td><td>Units</td><td>500</td><td>400</td><td>1000</td></tr>
<tr><td>Incomes</td><td>1000</td><td>800</td><td>2000</td></tr>
</table>

Ans. : HTML code is as follows :

```
<HTML>
<HEAD>
<TITLE> Sales Analysis </TITLE>
</HEAD>
<BODY>
<TABLE BORDER = "5" WIDTH = "100%" CELL SPACING = "15">
     <TR>
             <TD WIDTH = "40%" COLSPAN = "2"
                        ROWSPAN = "2"> </TD>
             <TD WIDTH = "60%" COLSPAN = "3"
                        ALIGN = "CENTER">
             <B> YEAR
             </B> </TD> </TR>
     <TR>
             <TD WIDTH = "20%" ALIGN = "CENTER">
                        <B> 1998 </B> </TD>
             <TD WIDTH = "20%" ALIGN = "CENTER">
                        <B> 1999 </B> </TD>
             <TD WIDTH = "20%" ALIGN = "CENTER">
                        <B> 2000 </B> </TD>
     </TR>
     <TR>
             <TD WIDTH = "20%" ROWSPAN = "2"
                        ALIGN = "CENTER">
                                <B> Sales </B> </TD>
             <TD WIDTH = "20%" ALIGN = "CENTER">
                                <B> Units </B> </TD>
             <TD WIDTH = "20%" ALIGN = "CENTER">
                                <B> 500 </B> </TD>
             <TD WIDTH = "20%" ALIGN = "CENTER">
                                400 </TD>
```

```
                <TD WIDTH = "20%" ALIGN = "CENTER">
                        100 </TD>

        </TR>
        <TR>
                <TD WIDTH = "20%" ALIGN = "CENTER">
                        <B> Income </B> </TD>
                <TD WIDTH = "20%" ALIGN = "CENTER">
                        1000 </TD>
                <TD WIDTH = "20%" ALIGN = "CENTER">
                        800 </TD>
                <TD WIDTH = "20%" ALIGN = "CENTER">
                        2000 </TD>

        </TR>
</TABLE>
</BODY>
</HTML>
```

Q. 39 Write a HTML code for a web page displaying local time and day. Make use of VB Script.

Ans. : HTML code :

```
        <HTML>
        <HEAD>
        <TITLE> Local time and day </TITLE>
        </HEAD>
        <BODY>
        <HI> Local time is .. </HI>
        <HR>
The local time is now
                <SCRIPT LANGUAGE = "VB SCRIPT">
Document Write Time ( ) & "on" & Month Name (Month (Now), False) & " " & Day
(Now) & ", " and Year (Now)
        </SCRIPT>
        </BODY>
        </HTML>
```

Q. 40 Write a HTML code with VB Script for a web page which greets "Good Morning" if time is 12.00 A.M. to 12.00 P.M. else greets "Good Afternoon". **(March 04, 19)**

Ans. : HTML code :

```
        <HTML>
        <HEAD>
        <TITLE> Greeting </TITLE>
        </HEAD>
```

```
<BODY>
<SCRIPT LANGUAGE = "VB SCRIPT">
        Function Greeting ()
                  If time ( )>#12:00 AM # And
        Time ( )<#12.00 PM # Then
                  greeting = "Good Morning"
        ELSE
                  greeting = "Good Afternoon"
        End If
                  End function
        Document. write greeting ( )
</SCRIPT>
<BR>
<H2> and welcome to web page </H2>
</BODY>
</HTML>
```

Q. 41 Write the extract output of the following HTML code with font specifications in brackets :

Ans. :

```
<HTML>
<body>
<h1> LIST OF BOOKS </h1> <hr>
<ul type="circle">
      <li> How to solve it By computer
      <li> HTML in Easy Steps
      <li> C++ Programming
</ul>
      <ol type = "A">
      <li> Microprocessor Programming
      <li> Networking Essentials
      <li> Microcontrollers.
</ol>
</body>
</html>
```

(March 2002, 07)

Ans. : Output is as follows with font specifications in brackets :

```
LIST OF BOOKS          ← (Text size h1 default font is used)
   o   How to solve it By computer
   o   HTML in Easy Steps           ← (Text size is default, Regular default font is used)
   o   C++ Programming
   A.  Microprocessor Programming     ← (Text size is default,
   B.  Networking Essentials            Regular default font is used)
   C.  Microcontrollers.
```

Q. 42 Write HTML code for a webpage displaying the following table. [Oct 2002]

Yesterday's Weather

City	High	Low	Wind
Mumbai	33	24	West
Pune	34	25	South
Latur	32	20	South

Ans. : HTML code is as follows :

```
<HTML>
    <HEAD>
        <TITLE>Weather Table</TITLE>
    </HEAD>
    <BODY>
        <TABLE BORDER = 2>
            <CAPTION>
                Yesterday's Weather
            </CAPTION>

        <TR>
            <TH>City</TH>
            <TH>High</TH>
            <TH>Low</TH>
            <TH>Wind</TH>
        </TR>
        <TR>
            <TD>Mumbai</TD>
            <TD align = "center">33</TD>
            <TD align = "center">24</TD>
            <TD align = "center">West</TD>
        </TR>
        <TR>
            <TD>Pune </TD>
            <TD align = "center">34</TD>
            <TD align = "center">25</TD>
            <TD align = "center">South</TD>
        </TR>
        <TR>
            <TD>Latur</TD>
            <TD align = "center">32</TD>
            <TD align = "center">20</TD>
            <TD align = "center">South</TD>
        </TR>
```

Q. 43 Write the exact output for the following code with font specifications in brackets.

```
<html>
<title> Introduction </title>
<body>
<h2> <b> Terms in computer </b> </h2>
<dl>
    <dt> Software
    <dd> <b> software </b> is a set of programs which is required to run the system
    <dt> Hardware
    <dd> The Electronic components used in the computer system is called as <b>
    Hardware </b>
</dl>
</body>
</html>
```

Ans. : Output is as follows with font specifications in bracket :

Introduction ← Title of the page is Introduction.

Terms in Computer : ← (Text size in h2, bold, default font is used)

Software :

Software is a set of programs which is required to run the system.

← (Text size default regular font is used)

Hardware :

The Electronic component used in the computer used in the computer system is called as Hardware.

Q. 44 Write the exact output of the following HTML code with font specifications in brackets : **(October 2003)**

```
<html>
<title> Introduction </title>
<body>
<h1> <b> COMPUTER SCIENCE </b> </h1>
<hr>
<u> SCHAUM'S OUTLINE SERIES < / u >
<hr>
<h5> SEYMOUR LIPSCHUTZ </h5>
</body>
</html>
```

Ans. :

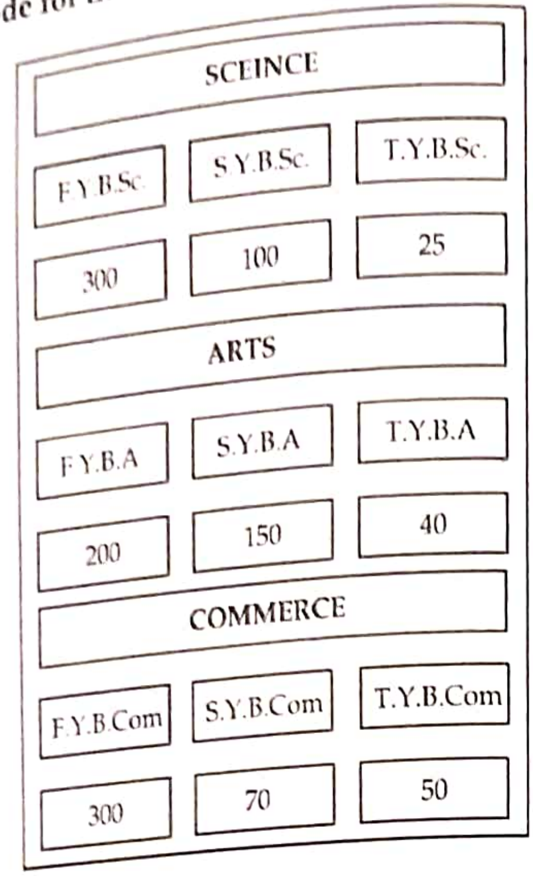
Introduction ← (Title of the page)

COMPUTER SCIENCE ← (text size h1 in bold, Default font is used)

SCHAUM' S OUTLINE SERIES ← (text size is default, Regular, Default font is used)

SEYMOUR LIPSCHUTZ ← (text size h5, Regular, Default font is used)

Q. 45 Write the HTML code for the following : (Mar.07, Oct.200



SCEINCE		
F.Y.B.Sc	S.Y.B.Sc.	T.Y.B.Sc.
300	100	25
ARTS		
F.Y.B.A	S.Y.B.A	T.Y.B.A
200	150	40
COMMERCE		
F.Y.B.Com	S.Y.B.Com	T.Y.B.Com
300	70	50

Ans. : HTML code is as follows :

```
<HTML>
    <BODY>
        <TABLE BORDER = "1" CELLPADDING = "20" CELLSPACING = "10">
        <TR>
            <TH COLSPAN = "3" align = "center"> SCIENCE </TR>
        <TR>
            <TD> FY BSc </TD>
            <TD> SY BSc </TD>
            <TD> TY BSc </TD>
        </TR>
        <TR> <TD ALIGN = "CENTER"> 300 </TD>
            <TD ALIGN = "CENTER"> 100 </TD>
            <TD ALIGN = "CENTER"> 25 </TD> </TR>
        <TR> <TH COLSPAN = 3> ARTS </TR>
        <TR> <TH> FY BA </TH>
            <TH> SY BA </TH>
            <TH> TY BA </TH> </TR>
        <TR> <TD ALIGN = "CENTER"> 200 </TD>
            <TD ALIGN = "CENTER"> 150 </TD>
            <TD ALIGN = "CENTER"> 40 </TD> </TR>
        <TR> <TH COLSPAN = 3> COMMERCE </TH> </TR>
        <TR> <TH> FY Bcom </TH>
             <TH> SY Bcom </TH>
             <TH> TY BCom </TH> </TR>
        <TR> <TD ALIGN = CENTER> 100 </TD>
             <TD ALIGN = CENTER> 70 </TD>
             <TD ALIGN = CENTER> 50 </TD> </TR>
        </TABLE>
    </BODY>
</HTML>
```

Q. 46 Write the HTML code for the following table : **(March 2004)**

		Year		
		1999	2000	2001
Sales	Units	300	750	1,200
	Income	Rs. 3,000	Rs. 7,500	Rs. 12,000

Ans. : The HTML code are as follows :

```
<HTML>
    <BODY>
        <TABLE BORDER = 1 CELLPADDING = 20>
        <TR>
                <TH ROWSPAN = "2" COLSPAN = "2">
                <TH COLSPAN = "3"> Year
        </TR>
                <TR>
                        <TH> 1999 </TH>
                        <TH> 2000 </TH>
                        <TH> 2001 </TH>
                </TR>
        <TR>
                <TH ROWSPAN = "2"> Sales
                <TH> Units </TH>
                        <TD> 300 </TD>
                        <TD> 750 </TD>
                        <TD> 1,200 <TD>
        </TR>
        <TR>
                <TH> Income </TH>
                        <TD> Rs. 3,000 </TD>
                        <TD> Rs. 7,500 </TD>
                        <TD> Rs. 12,000 </TD>
        </TR>
        </TABLE>
    </BODY>
</HTML>
```

Q. 47 Write the exact output for the following codes specifying RGB colours, output specifications in brackets : **(Oct. 2004)**

```
<html>
<head>
<title> computer shop </title>
</head>
<body>
<h3> <b> Title : XYZ Computers Ltd. </b> </h3>
<p> Address : Shakti Complex, Aurangabad </p>
<h1> <i> Dealers in : all types of peripherals </i> </h1>
</body>
</html>
```

Ans. : Output is as follows with font and RGB colors specification in bracket :

COMPUTER SHOP ← (Title of the page)

Title : XYZ Computers Ltd. ← (Text size in h3, bold. Default font is used default text colour)

Address : Shakti Complex, Aurgangabad. ← (Text size is default, regular, Default font is used. Default text color)

Dealers in : all types of peripherals. ← (Text size in h1, italic, Default font is used, Default text colour)

Q.48 Write HTML code for displaying a Web Page containing a six-celled table as shown below **(Oct. 2004)**

Sachin	Sourav	Laxman
First	Second	Third

Ans. : HTML code is as follows :

```
<HTML>
    <HEAD>
        <TITLE> Celled Table </TITLE>
    </HEAD>
    <BODY>
    <TABLE BORDER = 2 CELLSPACING = 50>
        <TR>
            <TD> Sachine </TD>
            <TD> Saurav </TD>
            <TD> Laxman </TD>
        </TR>
        <TR>
            <TD> First </TD>
            <TD> Second </TD>
            <TD> Third </TD>
        </TR>
    </TABLE>
    </BODY>
</HTML>
```

Q.49 Write the exact output of the following HTML code with font specifications in brackets : **(March 2005)**

```
    <html>
    <body>
    <h1 align = "center"> LIST OF TOPICS </h1>
    <hr>
    <ol>
    <li> <p align = "left"> Operating systems </p>
    <li> <p align = "left"> Data structures </p>
    <li> <p align = "left"> C++ programming </p>
    <li> <p align = "left"> HTML </p>
    </ol>
    <hr noshade>
    <ol>
    <li> <p align = "center"> COMPUTER SCIENCE </p>
    </ol>
    <hr>
    </body>
</html>
```

Ans. :

LIST OF TOPICS ← **(Text size in H1, regular, default font is used)**

↑ (Horizontal line with default width)

1. Operating systems
2. Data structures ← (Textsize regular, default font is used)
3. C++ programming
4. HTML

↑ (Horizontal line with reflected color)

1. COMPUTER SCIENCE ← (Textsize regular, default font is used)

↑ (Horizontal line with default width and size)

Q. 50 Write HTML code for the following : 2005, 2011, 2017

COMPUTER SCIENCE	PAPER-I	PAPER-II	TOTAL
	100	100	200

Ans. :

HTML code is as follows :

```
<HTML>
<HEAD>
    <TITLE> COMPUTER PAPAER ANALYSIS </TITLE>
</HEAD>
<BODY>
    <TABLE BORDER = 2 WIDTH = "100%"
                CELLSPACING = 15>
    <TR> <TD WIDTH = "25%" ROWSPAN = "2"
        ALIGN = "CENTER" >
        COMPUTER <BR> SCIENCE </TD>
    <TD WIDTH = "25%" ALIGN = "CENTER" >
        PAPER - I </TD>
    <TD WIDTH = "25%" ALIGN = "CENTER" >
        PAPER - II </TD>
    <TD WIDTH = "25%" ALIGN = "CENTER" >
        TOTAL </TD>
    </TR>
    <TR> <TD WIDTH = "25%" ALIGN = "CENTER" >
        100 </TD>
    <TD WIDTH = "25%" ALIGN = "CENTER" >
        100 </TD>
    <TD WIDTH = "25%" ALIGN = "CENTER" >
        200 </TD>
```

Q. 51 **Write an exact output of the following HTML code with font specifications in brackets :** **(Oct. 2005)**

```
<HTML>
     <BODY>
          <H1><CENTER>HSC SYLLABUS
          </CENTER></H1>
          <H3>PAPER I </H3>
          To know more about
          <P> Just click on here
          <ADDRESS>
          <BR> <B> Data structures </B>
          <BR> <B> GUI </B>
          <BR> <B> HTML </B>
          <BR> <B> C++ </B>
          <BR> <B> VB </B>
          </ADDRESS>
     </BODY>
</HTML>
```

Ans. :

HSC SYLLABUS ← (Text size H1)

PAPER I ← (Text size H3, regular font)

To know more about ← (Text size H3, regular font)

Just click on here ← (Text size regular)

Data Structure ← (Regular text size, Bold, unique font)

GUI

HTML

C++

VB

Q. 52 Write HTML code for web page displaying the following table : (Oct. 2005

Weather Forecast

City	Wind speed	From	To
Mumbai	39	South	West
Raigad	37	South	West
Panaji	42	East	South

Speed in Km/hr.

Ans. :

```
<HTML>
    <HEAD>
         <TITLE>Table</TITLE>
    </HEAD>
<BODY>
    <TABLE BORDER = 2>
         <CAPTION ALIGN = "TOP">
               <B> Weather Forecast </B>
         </CAPTION>
               <TR>
               <TH> City </TH>
               <TH> Wind speed </TH>
               <TH> From </TH>
               <TH> To </TH>
         </TR>
         <TR>
               <TD> Mumbai </TD>
               <TD ALIGN=CENTER> 39 </TD>
               <TD ALIGN=CENTER> South </TD>
               <TD ALIGN=CENTER> West </TD>
         </TR>
         <TR>
               <TD> Raigad </TD>
               <TD ALIGN=CENTER> 47 </TD>
               <TD ALIGN=CENTER> South </TD>
               <TD ALIGN=CENTER> West </TD>
         </TR>
         <TR>
               <TD> Panaji </TD>
               <TD ALIGN=CENTER> 42 </TD>
               <TD ALIGN=CENTER> East </TD>
               <TD ALIGN=CENTER> South </TD>
```

```
            </TR>
        </TABLE>
        <P> Speed in km/hr. </P>
    </BODY>
</HTML>
```

Q. 53 Write the HTML code for the following : **(March 2005)**

		Faculty		
		Arts	Science	Commerce
Students	Boys	100	400	500
	Girls	300	300	400

Ans. : The HTML code is as follows :

```
<HTML>
    <BODY>
        <TABLE BORDER = "3" CELLSPACING = "10">
        <TR>
        <TD ROWSPAN ="2" COLSPAN ="2">
        <TD COLSPAN = "3"> Faculty </TD>
        </TR>
        <TR>
                <TD> Arts </TD>
                <TD> Science </TD>
                <TD> Commerce </TD>
        </TR>
        <TR>
                <TD ROWSPAN ="2"> Students </TD>
                <TD> Boys </TD>
                <TD> 100 </TD>
                <TD> 400 </TD>
                <TD> 500 </TD>
        </TR>
        <TR>
                <TD> Girls </TD>
                <TD> 300 </TD>
                <TD> 300 </TD>
                <TD> 400 </TD>
        </TR>
    </TABLE>
    </BODY>
</HTML>
```

Q. 54 **Write the exact output of the following HTML code with font specification in brackets :**

```
<HTML>
    <TITLE> Introduction </TITLE>
    <BODY>
        <H1><B> Computer Science </B></H1>
        <HR>
        <U> E Balaguru Samy </U>
        <HR>
        <H4> Achyut S Godbole </H4>
    </BODY>
</HTML>
```

Ans. :

Introduction ← (Title of the page)

Computer Science ← (Text size h1 in bold default font is used)

<u>E Balaguru Samy</u> ← (Text size is default, Regular, default font is used underlined word)

Achyut S Godbole ← (Text size is h4, regular, default font is used)

Q. 55 **Write HTML code for the following :** **(Oct. 2006)**

Government of Maharashtra ← Text size in h2

<table>
<tr><td colspan="2">Education</td></tr>
<tr><td>PWD</td><td>ADMN</td></tr>
</table>

for details

feel free to approach us

PWD is a link available where clicking on PWD a webpage file 'Mahapwd.html' should be invoked. On clicking ADMN a webpage file 'admn.html' should be invoked. On clicking for details the user can write e-mail to mail-address 'free_free@hscboard.com'. On clicking feel free to approach us, user can invoke to website 'www.maharashtra.gov.in'.

Ans. :

```
<HTML>
<BODY>
```

```
<H2>Government of Maharashtra</H2>
<table border=1 cellspacing = "5">
<tr><th colspan = "2">
Education </th>
</tr> <TR>
</td> <a href = "Mahapwd.html">PWD
</a></td>
<td>
<a href = "admn.html"> ADMN
</a> </td>
</tr>
</table>
<P> <a href="free-free@hscboard.com"> for details</a>
<P>
<a href="www.maharashtra.gov.in"> feel free to approach us </a>
</BODY>
</HTML>
```

Q. 56 Write the exact output of the following HTML code with font specification in brackets : **(Oct. 2006)**

```
<html>
<title>Examination</title>
<body>
<h1><b>First semester Exam</b></h1>
<hr>
<u>MATHEMATICS</u>
<hr>
<u>STATISTICS</u>
<hr>
<u>BIOLOGY</u>
<hr>
<h5>MSEC BOARD</h5>
</body>
</html>
```

Ans. :

Examination ← Title of page

First semester Exam → bold with h1 size

<u>MATHEMATICS</u>

<u>STATISTICS</u>

<u>BIOLOGY</u>

MSEC BOARD ← size h5

Q. 57 Write HTML code for the following : (Oct. 2007)

Subjects

1.English (compulsory)

2.Second Languages ....

o Physics

● Chemistry

■ Maths

■ Biology

Ans :

HTML code

```
 <html>
 <head> </head>
 <BODY>
 <H1> <B> Subjects </B> </H1>
 <OL>
       <li> English (compalsory)
       <li> Second Language _ _ _ _ _
<UL>
       <LI type = "Circle"> Physics
<LI type = disc > Chemistry
<UL type = "Square">
       <LI> Maths
       <LI> Biology
</UL> </UL> </OL>
</BODY>
</HTML>
```

Q. 58 Write the exact output of following HTML code : (Oct. 2003, 2007, March 2004, 2006)

```
<html>
<head>
<Title> </Title>
</head>
<body>
<table border = "2" width = "50%">
<tr>
<td width = "50%" colspan = "2">
<h1 align = "center" > HSC Board Exams </h1>
</td>
</tr>
<tr>
<td width = "25%" align = "center"
```

```
<u>
Paper I </u> </td>
<td width = "25%" align = "center"> <u>
Paper II </u> </td> </tr>
<tr>
<td width = "25%
<P align = "center"> <i> 50 Marks </i> </td>
<td> width = "25%" align = "center">
<i> 50 Marks </i> </td>
</tr>
</table>
</body>
</html>
```

Ans. : Output of HTML code

HSC Board Exam	
Paper - I	Paper – II
50 Marks	*50 Marks*

Q. 59 Write the exact output of the following HTML code with font specifications in brackets : **(March 2008)**

```
<HTML>
<TITLE> INTRODUCTION </TITLE>
<BODY>
<H1> <B> PCMBEC </B> </H1>
<HR>
<U> NOBAL'S SERIES </U>
<H5> WELLKNOWN </H5>
</BODY>
</HTML>
```

Ans. :

Introduction (Title of page)

PCMBEC (Text size hl in bold default font is used)

NOBAL'S SERIES

WELLKNOWN (Text size h.5)

Q. 60 Write HTML code for the following : **(March 2008)**

		Year		
		2000	**2001**	**2002**
Sales	**Units**	500	1,000	1,500
	Income	Rs. 5,000	Rs. 10,000	Rs. 15,000

Ans : The HTML code as follows

```
        <HTML>
            <BODY>
               <TABLE BORDER = 1 CELLPADDING = 20>
               <TR>
               <TH ROWSPAN = "2" COLSPAN = "2"> </TH>
               <TH COLSPAN = "3" > Year </TH>
                   </TR>
                   <TR>
                   <TH> 2000 </TH>
                   <TH> 2001 </TH>
                   <TH> 2002 </TH>
               </TR>
            <TR>
               <TH ROWSPAN = "2" > Sales
               <TH> Units </TH>
                   <TD> 500 </TD>
                   <TD> 1,000 </TD>
                   <TD> 1,500 </TD>
               </TR>
               <TR>
                   <TH> Income </TH>
                   <TD> Rs. 5,000 </TD>
                   <TD> Rs. 10,000 </TD>
                   <TD> Rs. 15,000 </TD>
               </TR>
               </TABLE>
            </BODY>
            </HTML>
```

Q. 61 Write the exact output of the following HTML code: **(Oct. 2008)**

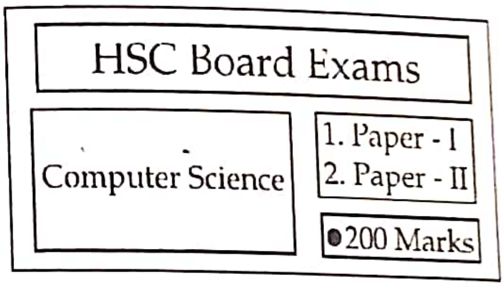
```
    <html>
    <head>
    <title> </title>
    </head>
    <body>
    <hl> Languages </hl>
    <ol>
        <li> English </li>
        <li> Second Languages ... </li>
        </ol>
  <ul>
```

```
    <li> Marathi </li>
    <ul>
    <li> Hindi </li>
    <ul>
    <li> French </li>
    </ul>
    <li> Sanskrit </li>
</ul>
</body>
</html>
```

Ans.

```
Languages                    ← Text size h1
1. English
2. Second Languages .....
● Marathi
     o Hindi
          ● French
     o Sanskrit
```

Q. 62 Write HTML coade for the following : **(Oct. 2008)**



<table>
<tr><td colspan="2">HSC Board Exams</td></tr>
<tr><td rowspan="2">Computer Science</td><td>1. Paper - I
2. Paper - II</td></tr>
<tr><td>● 200 Marks</td></tr>
</table>

Ans.

```
<html>
<head>
<title> Table Creation </title>
</head>
<body>
<table border = "5" width = "50%">
<tr>
            <td width = "100 %" colspan = "2">
            <h1 align = "centre"> HSC Board Exams </h1>
</tr>
<tr>
            <td width = "50 %" rowspan = "2">
            <p align = "center"><b>Computer Science </b></td>
            <td width =- "50 %" align = "center">
```

```
                <0l>
                        <li> paper I</li>
                        <li> paperII</li>
                </0l>
                </td>
    </tr>
    <tr>
                <td width = “50 %” aligh = “centre”>
                <ul>
                        <li><p align = “centre”>200 Marks </li>
                </ul>
                </td>
</tr>
</table>
</body>
</html>
```

Q. 63 Write HTMl code for the following output : **(March 2009)**

- ART
  - MARATHI
  - HINDI
  - ENGLISH
- COMMERCE
  - ACCOUNT
  - COSTING
  - AUDITING
- SCIENCE
  - PHYSICS
  - CHEMISTRY
  - MATHS
  - COMPUTER SC.

Ans :

```
<HTML>
<BODY>
<UL>
<LI>ART
    <UL>
```

```
        <LI> MARATHI
        <LI> HINDI
        <LI> ENGLISH
        </UL>
    <LI>COMMERCE
        <UL>
        <LI> ACCOUNT
        <LI> COSTING
        <LI> AUDITING
        </UL>
    <LI> SCIENCE
        <UL>
        <LI> PHYSICS
        <LI> CHEMISTRY
        <LI> MATHS
        <LI> COMPUTER SC.
        </UL>
    </UL>
    </BODY>
    </HTML>
```

Q. 64 Write the HTML Code for the following table : **(October. 2009)**

Ind 387	Yuvraj 183*
	Sehwag 83
Eng 238	Peterson 58
	Bopara 49
IND win I ODI by 149	

Ans :

```
<html>
<body>
<table border = "1">
<TR>
<TH Rowspan = "2"> IND 387 </TH>
<TD> Yuvraj 138 <sup> * </sup> </TD>
</TR>
<TR>
<TD> Sehwag 83 </TD>
</TR>
<TR>
<TH Rowspan = "2"> Eng 238 </TH>
<TD> Peterson 58 </TD>
</TR>
```

```
<TR>
<TD> Bopara 49 </TD>
</TR>
<TR>
<TD Colspan = "2"> IND win I <sup> st </sup> ODI by 149
</TR>
</table>
</body>
</html>
```

Q.65 Write the HTML Code for the following table : (Oct.2009)

Year	Students		
	Boys	Girls	Total
2006	55	75	130
2007	75	95	170

Record

Ans :

```
<html>
<body>
<table border = "1">
<Caption align="bottom"> Record</caption>
<TR>
<TH rowspan = 2>year </TH>
<TH colspan = 3>student </TH>
</TR>
<TR>
<TH> Boys
<TH> Girls
<TH> Total
</TR>
<TR>
<TD>2006
<TD>55
<TD>75
<TD>130
</TR>
<TR>
<TD>2007
<TD> 75
<TD>95
<TD>170
</TR>
```

```
</table>
</body>
</html>
```

Q. 66 Write exact output of the following HTML Code with font specifications in brackets : **(March 2010)**

```
<html>
<body>
<h1> <u> Network Connectivity Devices </u></h1>
<ul>
<li> Modem
<li> Hub
<ul>
<li> Repeater
<li> Router
</ul>
</ul>
</body>
</html>
```

Ans : Output is as follows with font specifications in brackets :

```
Network Connectivity Devices – (Text size in h1,
●   Modem               Regular default
●   Hub                 font is used)
    O   Repeater        (Text size is default
    O   Router          Regular font is used)
```

Q. 67 Write HTML Code for the following : **(Mar.2010)**

		No. of Books Purchased	
		F.Y.J.C.	S.Y.J.C.
Year	2004	1200	1300
	2005	1250	1400

Ans :

```
<HTML>
    <BODY>
        <TABLE border = 1 Cellpadding = 20>
        <TR>
            <TH Rowspan = 2 Colspan = 2> </TH>
            <TH Colspan = 2> No. of Books Purchased </TH>
        </TR>
        <TR>
<TH> F.Y.J.C. </TH>
<TH> S.Y.J.C. </TH>
```

```
        </TR>
        <TR>
        <TH Rowspan = 2> Year </TH>
        <TH> 2004 </TH>
        <TD> 1200 </TD>
        <TD> 1300 </TD>
       </TR>
       <TR>
          <TH> 2005 </TH>
          <TD> 1250 </TD>
          <TD> 1400 </TD>
    </TR>
    </TABLE>
    </BODY>
    </HTML>
```

Q. 68 Write exact output of the following HTML code with font specifications in brackets:

```
<html>
<body>
<h1> Terms used in Networking </h1>
<hr>
<ul type = "circle">
<li> Bandwidth
<li> Attenuation
<li> Electromagnetic Interference
</ul>
<ol type = "a">
<li> Topology
<li> Ethernet
<li> Protocol
</ol>
</body>
</html>
```

Ans :
```
Terms used in Networking        ← (Text size in h1, regular, default font is used)
  O Bandwidth                      (Horizontal line with default width)
  O Attenuation
  O Electromagenetic Interference ← (Text size regular, default font is used)
  a. Topology
  b. Ethernet           ← (Ordered list with text size regular, default font issued)
  c. Protocol
```

Q.69 Write HTML code's output for the following : **(Oct. 2010 ; March 2020)**

```
<html>
<body>
<hl> XII RESULT</hl>
<table border = "1" cellspacing = "10">
<tr>
<th colspan = "3" > <u> STREAM <N> </th>
</tr>
<tr>
<td> <a href = " SCIENCE. html" > SCIENCE </a> </td>
<td> <a href = " COMMERCE.html"> COMMERCE </a> </td>
<td> <a href = " ART. html"> ART </a> </td> </tr>
</table>
</body>
</html>
```

Ans. :

XII RESULT (Text size h1)

STREAM		
SCIENCE	COMMERCE	ARTS

SCIENCE is a link available where clicking on SCIENCE a web page file "SCIENCE.html" should be invoked, on cliking COMMERCE a web page file "COMMERCE.html" should be invoked and on clicking ART a web page file "ART.html" should be invoked.

Q.70 Write a HTML Code for the following output : **(March 2011) 5**

COMPUTER DEVICES ← (Text size h2, align Center)

- INPUT DEVICES

  1) Key Board

  2) Mouse

- STORAGE DEVICES

  1) Hard Disk

  2) Floopy Disk

  3) Compact Disk

- OUTPUT DEVICES

  1) Screen

  2) Printer

Ans :

```
< HTML >
    <Head>
    <title> List </ title>
</ Head>
```

```
<Body> <H2 ALIGN = " center"> COM
<UL>
      <LI> INPUT DEVICES
      <OL>
      <LI> Keyboard
      <LI> Mouse
      </OL>
<LI> STORAGE DEVICES
<OL>
      <LI> Hard Disk
      <LI> Floopy Disk
      <LI> Compact Disk
</OL>
      <LI> OUTPUT DEVICES
<OL>
      <LI> Screen
      <LI> Printer
</OL>
</UL>
</Body>
</HTML >
```

Q. 71 **Write the HTML Code for the following :** **(Oct. 2011)**

V.I.P. SALES		
STATIONARY	ERASER	5
	PENCIL	7
	PEN	10
	BOOK	22
TOTAL		44

Ans.:

```
<html>
<body>
      <table border = 2>
      <tr>
            <th colspan = 3>V.I.P. SALES</th>
      </tr>
      <tr>
            <th rowspan = 4>STATIONARY</th>
            <td>ERASER</td>
            <td>5</td>
```

```
    </tr>
    <tr>
            <td>PENCIL</td>
            <td>7</td>

    </tr>
    <tr>
            <td>PEN</td>
            <td>10</td>

    </tr>
    <tr>
            <td>BOOK</td>
            <td>22</td>

    </tr>
    <tr>
            <td.colspan = 2>TOTAL</td>
            <td>44</td>

    </tr>
    </table>
</body>
</html>
```

Q. 72 Write HTML Code 'o display the following nested list : **(Oct. 2011, 5 Marks)**

- College
    - I. Arts
        - (a) History
        - (b) Politics
        - (c) Languages
            - (i) English
            - (ii) Marathi
    - II. Science
        - (a) Physics
        - (b) Chemistry
        - (c) Biology
    - III. Commerce
        - ■ Accounts

Ans. :

```
<html>
<body>
    <ul type = disc>
          <li>College>
                <ol type = I>
                <li>Arts
                        <ol type = a>
                              <li>History
                              <li>Politics
                              <li>Languages
                              <ol type = i>
                                    <li>English
                                    <li>Marathi

                              </ol>

                        </ol>

                <li>Science
                        <ol type = a>
                              <li>Physics
                              <li>Chemistry
                              <li>Biology

                        </ol>

                <li>Commerce
                        <ul type = square>
                        <li>Accounts
                        </ul>

                </ol>

      </ul>
   </body>
   </html>
```

Q. 73 **Write exact output of the following HTML Code with font specifications in brackets:** **(March 2012, 5 Marks)**

```
<HTML>
<HEAD>
<TITLE>Computer Step</TITLE>
</HEAD>
<BODY>
<H2><U>Title : Megastar Company Ltd.</U></H2>
<P>Address : Surya Complex, Delhi</P>
<H4><B>Deals In : </B></H4>
<UL>
<LI>Software
<LI>Hardware
<LI>Peripherals
</UL>
</BODY>
</HTML>
```

Ans.:

Computer Step ← Title of page

Title : Megastar Company Ltd. ← text size H2

Address : Surya Complex, Delhi ← text size H4, Bold

Deals In :

- Software
- Hardware
- Peripherals

Q.74 Write HTML Code for the following : **(March 2012, 5 Marks)**

Roll No.	Name	Marks		Total
		CS1	CS2	
1.	RINA	75	70	145
2.	RONAK	80	90	170
3.	TINA	70	85	155

Ans.:

```
<html>
<body>
    <table border = "1" align = "center">
    <tr>
        <th rowspan = "2">Roll No.</th>
        <th rowspan = "2">Name.</th>
        <th colspan = "2">Marks</th>
        <th rowspan = "2">Total</th>
    </tr>
    <tr>
        <th>CS1</th>
        <th>CS2</th>
    </tr>
    <tr>
        <td>1
        <td>RINA
        <td>75
        <td>70
        <td>145
    </tr>
    <tr>
        <td>2
        <td>RONAK
        <td>80
        <td>90
```

```
            <td>170
        </tr>
        <tr>
            <td>3
            <td>TINA
            <td>70
            <td>85
            <td>155
        </tr>
        </table>
    </body>
    </html>
```

Q. 75 **Write a HTML Code to display the following output: :** **(Oct. 2012, 5 Marks)**

List of Books ← (Text size hl, default font is used)

O How to solve it By computer
O HTML in Easy steps
O C++ Programming
A. Microprocessor Programming
B. Networking Essentials
C. Microcontrollers

← Text size is default

```
<html>
<body>
<h1>List of Books </hl>
<UL type = "circle">
      <li> How to slove it By Computer
      <li> HTML in Esy steps
      <li>c++ Programming
      <ol type ="A">
      <li> Microprocessor Programming
      <li> Networking Essentials
      <li> Microcontrollers
</ol>
</UL>
</body>
</html>
```

Q. 76 **Write exact output of the following HTML. Code with font specifications in brackets :** **(March 2013, 5 Marks)**

```
<html>
<body>
<h1 aligh = "centre"> H.S.C BOARD EXAM </h1>
```

```
<p align = "centre"> <i> SUBJECT : COMPUTER SCIENCE </i> </p>
<p align = "centre"> <b> THEORY AND PRACTICAL EXAMS </b> </p>
<p align = "centre"> <u> TOTAL </u> 200 MARKS </p>
<body>
</html>
```

Ans. :

H.S.C. BOARD EXAMS ← (Text size H1)

SUBJECT COMPUTER SCIENCE ← (Italic text)

THEORY AND PRACTICAL EXAMS ← (BOLD text)

<u>Total</u> 200 MARKS

Q.77 Write HTML codes for the following **(March 2013, 5 Marks)**

HTML COLOUR CODE IN HEXA-DECIMAL	
BLACK	WHITE
# 000000	# FFFFFF

Ans. :

```
<html>
<table border = "1" cellspacing = "1" width = "50%">
<tr>
    <td width = "50%" colspan = "2">< p align = "center">
    HTML COLOUR CODE <br> IN HEXA DECIMAL </td>
</tr>
<tr>
    <td width = "25%">< p align = "center"> BLACK </td>
    <td width = "25%">< p align = "center"> WHITE </td>
</tr>
<tr>
    <td width = "25%">< p align = "center"> #000000 </td>
    <td width = "25%">< p align = "center"> FFFFFF </td>
</tr>
</table>
</body>
</html>
```

Q.78 Write the exact output of the following HTML Code with font specifications in brackets : **(Oct. 2013, 5 Marks)**

```
<html>
<body>
<dl>
<h2><b> Terms related to computer </b></h2>
```

```
<dl>
<dt> Protocols
<dd><b> Protocols </b> are rules by which communicates
<dt> Byte
<dd><b> Byte </b> is a group of 8 bit
<dl>
</body>
</html>
```

Ans. :

```
        Terms related to computer    ← bold, size h2
    Protocols
        Protocols are rules by which communicates.
    Byte
        Byte is a group of 8 bit.
```

Q. 79 Write HTML code for the following table : **(Oct. 2013, 5 Marks)**

Library Books		
Section	Reference	Text Book
	2000	4000

Ans. :

```
<html>
<body>
<table border = "1">
<tr>
    <th colspan = "3"> Library Books </th>
</tr>
<tr>
    <th rowspan = "2"> section </th>
    <th> Reference </th>
    <th> Text book </th>
</tr>
<tr>
    <th> 2000 </th>
    <th> 4000 </th>
</tr>
</table>
</body>
</html>
```

Q. 80 Write exact output of the following HTML code with font specifications in bracket : **(March 2014, 5 Marks)**

```
<HTML>
<BODY>
<H2> My COUNTRY </H2>
<HR> <BR>
<H1 ALIGN = "CENTER"> INDIA
        </H1> <BR>

<UL> IS
        <LI> GREAT
        <LI> BEAUTIFUL
        <LI> LOVING

</UL>
Here people care for each other
</BODY>
</HTML>
```

Ans. :

MY COUNTRY ←size H2

INDIA ← centre and size H1

IS

- GREAT
- BEAUTIFUL
- LOVING

Here people care for each other

Q. 81 Write an HTML code to display the following table on Web page : **(March 2014, 5 Marks)**

Record

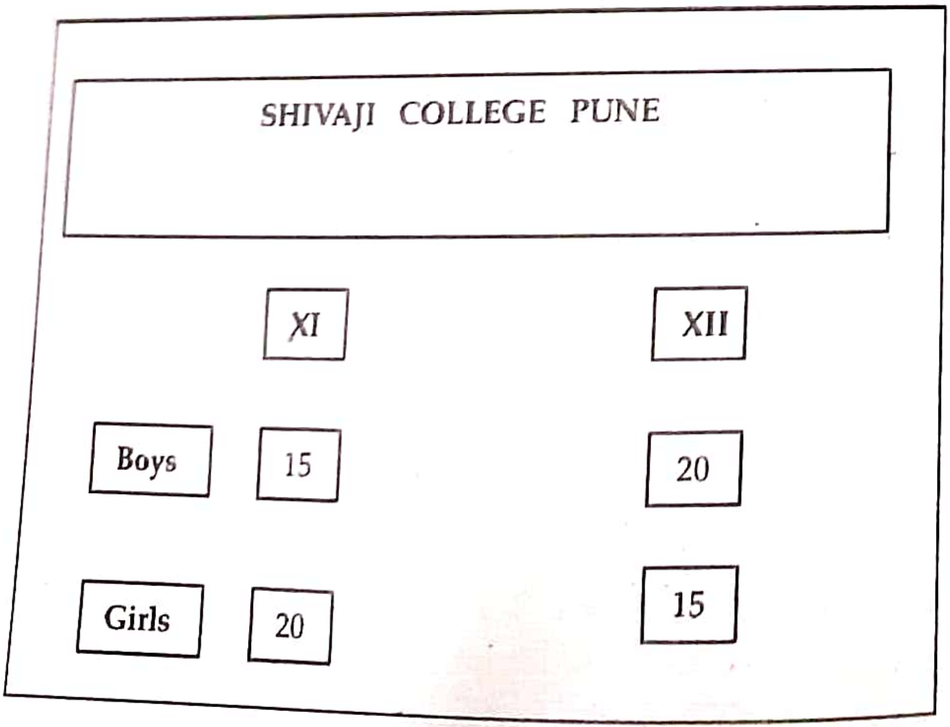
Year	Students		
	Boys	Girls	Total
2007	519	710	1229
2008	800	925	1725

Ans. :

```
<html>
<body>
<table border="1">
<caption> Record </caption>
```

```
      <tr>
            <th rowspan="2"> year</th>
            <th colspan="3"> students</th>
      </tr>
      <tr>
            <th>Boys</th>
            <th>Girls</th>
            <th>total</th>
      </tr>
      <tr>
            <th>2007</th>
            <td>519</td>
            <td>710 </td>
            <td>1229</td>
      </tr>
      <tr>
            <th>2008 </th>
            <td>800</td>
            <td>925</td>
            <td>1725</td>
      </tr>
</table>
</body>
</html>
```

Q. 82 Write HTML code to display following output : **(Oct. 2014, 5 Marks)**



SHIVAJI COLLEGE PUNE

	XI	XII
Boys	15	20
Girls	20	15

Ans. :

```
<HTML>
<BODY>
<TABLE BORDER = "2" CELLSPACING = "20" CELLSPADDING = "5">
<TR>
<TH COLSPAN = "3"> SHIVAJI COLLEGE PUNE </TH>
</TR>
<TR>
<TH>                      </TH>
<TH>          XI          </TH>
<TH>          XII         </TH>
</TR>
<TR>
<TH>          Boys        </TH>
<TD>          15          </TD>
<TD>          20          </TD>
</TR>
<TR>
<TH>          Girls       </TH>
<TD>          20          </TD>
<TD>          15          </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Q.83 Write HTML code to display following output : **(March 2015)**

```
<HTML>
    <HEAD> <TITLE> abc </TITLE> </HEAD>
    <body>
    <H1 align = "center> KBC Restaurant </H1>
    <TABLE border = 2>
<TR>        <TH Rowspan = "2">
    <Font size = "5"> Pawbhaji </font>
    </TH>
    <TH> with cheese </TH>
    <TH> 35
    </TR>
            <TR>
                    <TH> without cheese
                    <TH> 25
    <TR>
```

```
        </TABLE>
    </Body>
    </HTML>
```

Ans : abc Title

KBC Restaurant

Pawbhaji	With Cheese	35
	Without Cheese	25

Q. 84 Write a program in HTML for the following output **(March 2015, 5 Marks)**

(i) Arts
 A. History
 B. Geography

(ii) Science
 I. Computer science
 II. Physics

(iii) Commerce
 O English
 O Accounts

Ans. :

```
    <HTML>
    <BODY>
    <OL Type = "i">
        <LI>Arts
        <OL Type = "A">
            <LI> History
            <LI> Geography
       </OL >
       <LI> Science
       <OL Type =" I" >
            <LI> Computer Science
            <LI> Physics
      </OL>
      <LI> Commerce
      <UL Type= "Circle">
           <LI> English
           <LI> Accounts
</UL>
</OL>
<BODY>
</HTML>
```

Q.85 **Write output of following HTML code :** **(Oct. 2015, 5 Marks)**

```
<HTML>
     <HEAD>
          <TITLE> ABC </TITLE>
     </HEAD>
     <BODY>
          <HR>
          <CENTER> COMPUTER SCIENCE </CENTER>
  <HR>
     <ol>
          <LI> Operating System
          <LI> Data Structure
          <LI> C++
     </OL>
  </BODY>
  </HTML>
```

Ans. :

ABC ← Title

COMPUTER SCIENCE

---

1. Operating system
2. Data structure
3. C++

Q.86 **Write HTML Code to display following output :** **(Oct. 2015, 5 Marks)**

<table>
<tr><td colspan="2">Tuesday</td></tr>
<tr><td>March</td><td rowspan="2">18</td></tr>
<tr><td>2006</td></tr>
<tr><td>← Prev.</td><td>Next →</td></tr>
</table>

Ans. :

```
<HTML>
     <TABLE CELLSPACING = "2" BORDER = "2">
     <TR>
          <TD Colspan = "2" Align = "Center"> TUESDAY </TD>
     </TR>
     <TR>
     <TD Align = "Center"> March </TD>
     <TD Rowsapn = "2" Align = "Center"> <H1> 18 </H1> </TD>
     </TR>
     <TR>
          <TD Align = "Center"> 2006 </TD>
```

```
        </TR>
        <TR>
              <TD Align = "Center">← Prev. </TD>
              <TD Align = "Center"> Next→ </TD>
        </TR>
        </TABLE>
        </HTML>
```

Q. 87 Write an HTML code for following : **(March 2016, 5 Marks)**

Year	Students		
	Boys	Girls	Total
2004	25	30	55
2005	80	25	105

Record

Ans. :

```
 <HTML>
 <BODY>
 <TABLE BORDER = "1">
<CAPTION ALIGN = "BOTTEM"> RECORD </CAPTION>
     <TR>
     <TH > Year
     <TH COLSPAN = "3"> Students
     </TR>
     <TR>
         <TH>          </TH>
         <TH> Boys </TH>
         <TH> Girls </TH>
         <TH> Total </TH>
   </TR>
   <TR>
        <TD> 2004 </TD>
        <TD> 25 </TD>
        <TD> 30 </TD>
        <TD> 55 </TD>
  </TR>
  <TR>
       <TD> 2005 </TD>
       <TD> 25 </TD>
       <TD> 80 </TD>
       <TD> 105 </TD>
 </TR>
 </TABLE>
</HTML>
```

Q. 88 **Write an HTML code of following output :** **(March 2016, 5 Marks)**

COLLEGE ← h1 and center

Principal

- Vice Principal
- Professors
- Non-teaching Staff

For more details click here

(The words click here act or hyperlink to next page whose address is "C:\My Documents\A1.HTML")

Ans. :

```
<HTML>
<BODY>
<H1 ALIGN = "CENTER"> COLLEGE </H1>
   <UL> Principal
   <LI> Vice principal
   <LI> Professors
   <LI> Non teaching staff
   </UL>
   For more details
   <A href = "c:\My documents \ Al.HTML"> Click here </A>
   </BODY>
   </HTML>
```

Q. 89 **Write HTML code for following output :** **(July 2017)**

World Cup T20		
Group A	IND	PAK
	AUS	ENG
GROUP B	WI	NZ
	SA	SL

Ans. :

```
            <HTML>
   <HEAD>
   <TITLE>TABLE</TITLE>
   </HEAD>
   <BODY>
   <TABLE BORDER="1" WIDTH=25%>
   <TR>
       <TH COLSPAN="3" ALIGN="CENTER">World Cup T20</TH>
   </TR>
   <TR>
       <TD ROWSPAN="2" ALIGN="CENTER">Group A</TD>
       <TD ALIGN="CENTER"> IND </TD>
       <TD ALIGN="CENTER">PAK </TD>
   </TR>
   <TR>
```

```
        <TD ALIGN="CENTER"> AUS </TD>
        <TD ALIGN="CENTER"> ENG </TD>
    </TR>
    <TR>
        <TD ROWSPAN="2" ALIGN="CENTER">Group B</TH>
        <TD ALIGN="CENTER"> WI </TD>
        <TD ALIGN="CENTER"> NZ </TD>
    </TR>
    <TR>
        <TD ALIGN="CENTER"> SA </TD>
        <TD ALIGN="CENTER"> SL </TD>
    </TR>
        </TABLE>
        </BODY>
        </HTML>
```

Q. 90 Write HTML code for the following output : **(July 2017)**

□ Computer Science

● Paper - I

(i) C++ language

(ii) HTML

(iii) OS

(iv) DS

○ Paper - II

(1) Microprocessor

(2) Microcontroller

(3) X86 Processors

(4) Networking

Ans. :

```
<HTML>
<HEAD>
<TITLE>LIST</TITLE>
</HEAD>
<BODY>
<UL TYPE="SQUARE">
<LI> Computer Science</LI>
</UL>
<UL TYPE="DISC">
<LI> Paper-I </LI>
        <OL TYPE="i">
        <LI>C++ language<N>
        <LI>HTML</LI>
        <LI>OS</LI>
        <LI>DS</LI>
        </OL>
</UL>
<UL TYPE="CIRCLE">
```

```
<LI>Paper-II</LI>
        <OL TYPE="1">
        <1.1>Microprocessor</U>
        <LI>Microcontroller</LI>
        <LI>X86 Processors<N>
        <LI>Networking<N>
        </OL>

</UL>
</BODY>
</HTML>
```

.91 **Write HTML code for following output :** **(March 2018)**

Cricket Analysis

Country	Played	Won	Lose
INDIA	30	27	03
PAKISTAN	30	03	27

Ans. :

```
<HTML>
<HEAD>
    <TITLE> TABLE </TITLE>
</HEAD>
<BODY>
<TABLE BORDER="1" WIDTH="25%">
<CAPTION> <B> Cricket Analysis </B> </CAPTION>
<TR>
    <TH>Country</TH>
    <TH> Played </TH>
    <TH> Won </TH>
    <TH> Lose </TH>
</TR>
<TR>
    <TD> INDIA </TD>
    <TD align="center"> 30 </TD>
    <TD align="center"> 27 </TD>
    <TD align="center"> 03 </TD>
</TR>
<TR>
    <TD> PAKISTAN </TD>
    <TD> <align="center"> 30 </TD>
    <TD> <align="center"> 03 </TD>
    <TD> <align="center"> 27 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Q. 92 Write the output of the following 2015

```
<html>.
    <body>
        <UL type = "circle">
            < Li > One
            <Li> Two
            <Li> Three
        <UL type = "square'>
            <Li> Monday
            <Li> Tuesday
            <Li> Wednesday

        </UL>
        </UL>
    </body>
</html>
```

Ans. :

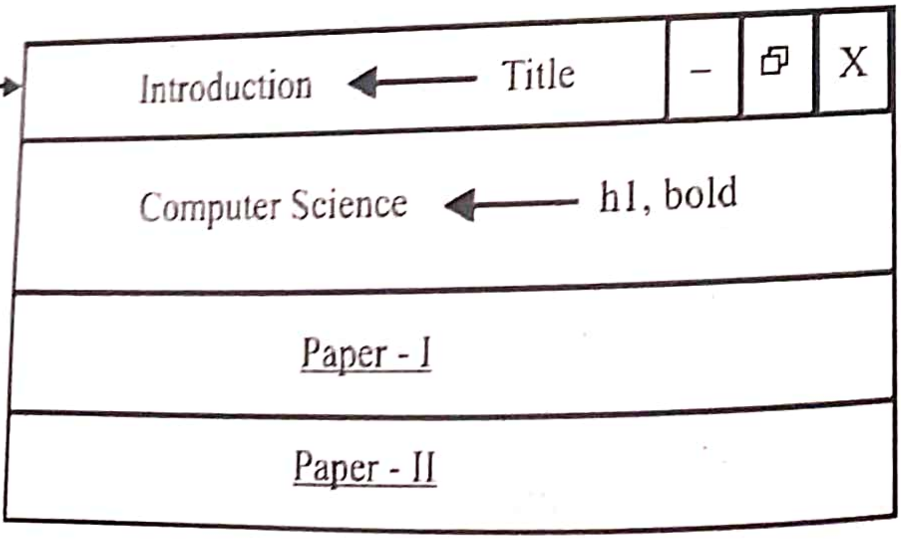
```
o   One
o   Two
o   Three
    ■   Monday
    ■   Tuesday
    ■   Wednesday
```

Q. 93 Write the exact output of the following HTML code with font specification in bracket. (March 2017, 5)

```
<html>
<title> Introduction </title>
<body>
<h1> <b> Computer Science </b> </h1>
<hr>
<u> Paper-I </u>
<hr>
<u> Paper-II </u>
</body>
</html>
```

Ans. :

Output Window :



Introduction ← Title

Computer Science ← h1, bold

Paper - I

Paper - II

Write the exact out ...g HTML code with font specifications in brackets. **(July 2016)**

```
<html>
<body>
<ul>
    <li> <i> This is in Italic form </i> </li>
    <li> <u> This is in underlined text </u> </li>
    <li> <b> This is in BOLD Text </b> </li>

<ul>
    <li> This a List </li>
    <li> ABC </li>
    <li> XYZ </li>

</ul>
</ul>
</body>
</html>
```

s.:

s is in *Italic form.*

is is <u>underlined text.</u>

is is **BOLD text.**

- This is a list
- ABC
- XYZ

.95 **Write the HTML code for the following :** **(July 2016)**

Total Marks	
Theory	Practical
100	100

Ans. :

```
<html>
<body>
<table border=1 cellspacing="10">
<tr>
    <td colspan="2" align="center">TOTAL MARKS</td>
</tr>
<tr>
    <td>THEORY</td>
    <td>PRACTICAL</td>
    </tr>
    <tr>
        <td align="cener">100</td>
        <td align="cener">100</td>

    </TR>
    </table>
    </body>
    </html>
```

Q. 96 Write the exact output of the following HTML code : **(July 2018)**

```
<HTML>
<BODY>
<h1> <U> Services of
Operating System </U> </H1>
<UL Type = "square">
<LI> Information Management
</LI>
<UL Type = "circle">
<LI> File system </LI>
<OL>
<LI> Tape based system
<LI> Disk based system
</OL>
</UL>
<LI> Process Management </LI>
<LI> Memory Management </LI>
</UL>
</BODY>
</HTML>
```

Ans. :

```
Service of Operating System

■   Information Management
      o     File System
               1.     Tape based system
               2.     Disk based system
■   Process Management
■   Memory Management
```

Q. 97 Write a code in HTML for following table : **(July 2018)**

RESULT	
JEE	CET

If any enquiry click here.

Hint :

(i) JEE is link to open a WebPage JResult.html

(ii) CET is link to open a Webpage CResult.html

(iii) Click here is a link to open website www.jeemain.nic.in

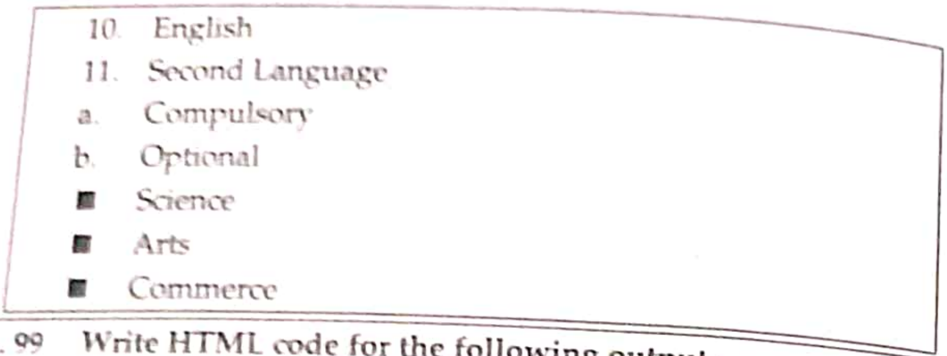
Ans. :

```
<html>
<body>
<table border="1" cellpadding=20 width="20%">
<tr>        <th colspan="2">RESULT</th>

</tr>
<tr>        <th><a href="JResult.html">JEE</a></th>
            <th><a href="CResult.html">CET</a></th>

</tr>
</table>
<br>
If any enquiry <a href="www.jeemain.nic.in"><b>click
here.</b></a>
</body>
</html>
```

Q. 98 Write exact output of the following HTML code : **(March 2019)**

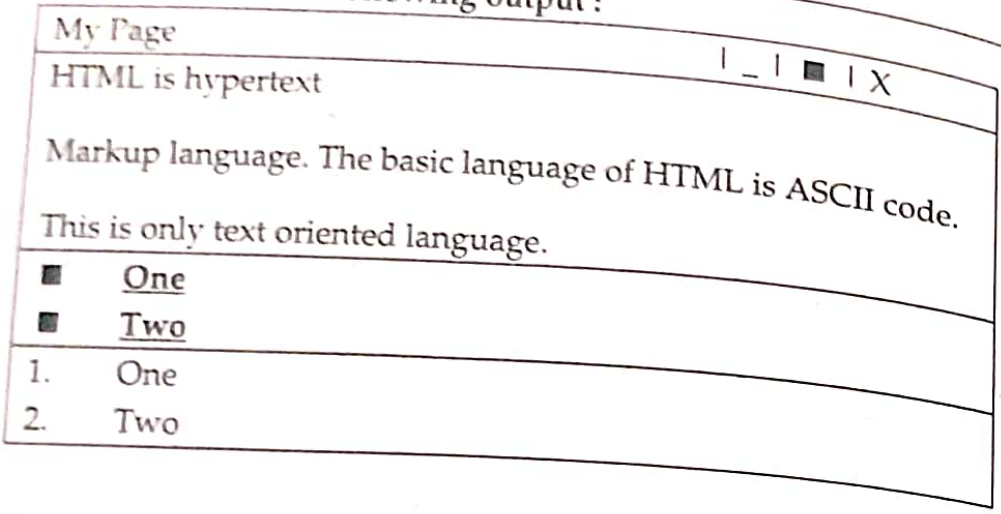
```
<HTML>
    <BODY>
        <OL start = "10">
                <li> English
                <li> Second language
        </OL>
        <OL Type = "a">
                <li> Compulsory
                <li> Optional
        </OL>
        <UL type = "Square">
                <Li> Science
                <Li> Arts
                <Li> Commerce
        </UL>
        </BODY>
  </HTML>
```

Ans. :



10. English
11. Second Language
a. Compulsory
b. Optional
■ Science
■ Arts
■ Commerce

Q. 99 Write HTML code for the following output : **(March 2019)**



HTML is hypertext

Markup language. The basic language of HTML is ASCII code.

This is only text oriented language.

■ **One**
■ **Two**

1. One
2. Two

Ans. :

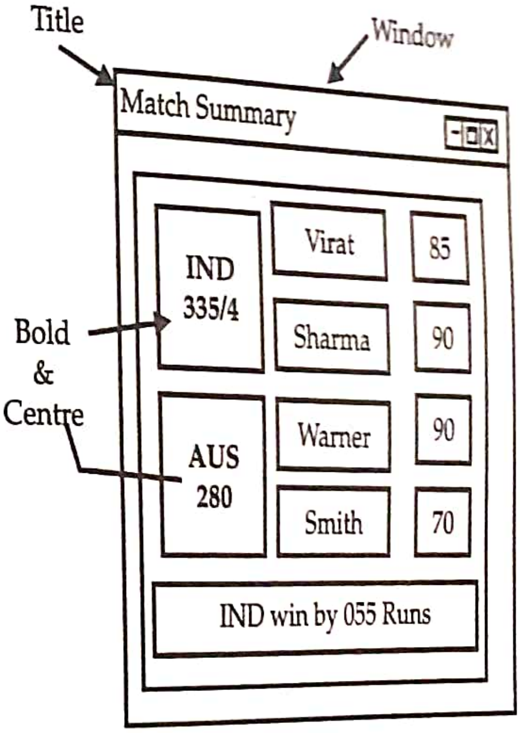
```
 <HTML>
 <HEAD>
       <TITLE> My Page</TITLE>
 </HEAD>
 <BODY>
 <P> HTML is a hypertext </P>
 <P>Markup, language. The basic language of HTML is ASCII code. The start tag and
 end tag.
 </P>
 <P> This is only text oriented language.
 </P>
 <HR>
 <B> <U> <UL TYPE="SQUARE">
                <LI>One</LI>
                <LI>Two</LI>
                </UL>
</U> </B>
<HR>
<OL type="1">
                <LI>One</LI>
```

```
        <LI>Two</LI>
</OL>
</BODY>
</HTML>
```

Q. 100 Write the exact output of the following HTML code : **(July 2019)**

```
<HTML>
<HEAD>
<TITLE> Match summary </TITLE>
</HEAD>
<BODY>
<Table border = "1" cellspacing = "15" cellpadding = "15">
<TR>
<TH Rowspan = "2"> IND <br>
    335/4 </TH>
<TD> Virat </TD>
<TD> 185 </TD>
</TR>
<TR>
<TD> Sharma </TD>
<TD> 90 </TD>
</TR>
<TR>
<TH Rowspan = "2"> AUS <br>
    280 </TD>
<TD> Warner </TD>
<TD> 90 </TD>
</TR>
<TR>
<TD> Smith </TD>
<TD> 70 </TD>
</TR>
<TR>
<TR> <TD Colspan = "3"> IND win
    by 055 Runs </TD>
</TR>
</Table>
</Body>
</HTML>
```

Ans. :



Match Summary

IND 335/4	Virat	85
	Sharma	90
AUS 280	Warner	90
	Smith	70
IND win by 055 Runs		

Fig. 1

Q. 101 Write the HTML Code for following output : **(July 2019)**

(1) Computer Science Theory
   (i) Paper 1 – 50 Marks
   (ii) Paper 2 – 50 Marks
(2) Computer Science Practicals
   (i) Paper 1 – 50 Marks
   (ii) Paper 2 – 50 Marks

Ans. :

```
<HTML>
<BODY>
<OL Type = "1">
    <LI> Computer Science Theory </LI>
    <OL Type = "i">
                <LI> Paper 1 – 50 Marks </LI>
                <LI> Paper 2 – 50 Marks </LI>

    </OL>
    <LI> Computer Science Practical </LI>
    <OL Type = "i">
                <LI> Paper 1 – 50 Marks </LI>
                <LI> Paper 2 – 50 Marks </LI>

  </OL>
</OL>
</BODY>
</HTML>
```

Q. 102 Write a code in HTML for following table : **(March 2020)**

Subject		Paper- I	Paper- II
Computer	Theory	50	50
Science	Practical	50	50

HSC Exam Scheme

Ans. :

```
<HTML>
<BODY>
<TABLE BORDER="2">
<CAPTION ALIGN="BOTTOM">HSC Exam Scheme</CAPTION>
    <TR>
        <TH COLSPAN="2"> Subject </TH>
        <TH> Paper-I</TH>
        <TH> Paper-II</TH>
```

```
</TR>
<TR>
<TD> Computer </TD>
<TD ALIGN="CENTER"> Theory </TD>
<TD ALIGN="CENTER"> 50 </TD>
<TD ALIGN="CENTER"> 50 </TD>
</TR>
<TR>
<TD> Science </TD>
<TD ALIGN="CENTER"> Practicals </TD>
<TD ALIGN="CENTER"> 50 </TD>
<TD ALIGN="CENTER"> 50 </TD>
</TR>
BODY>
HTML>
```

.103 **Select the correct alternative and rewrite the following.**

—— tag is used for superscript in HTML. **(Oct. 2002, 09, 11)**

(i) <SUPER> (ii) <SUP> (ii) <UP> (iv) <SUPERSCRIPT>

ns. : (ii) <SUP>

HTML stands for ——. **(Oct. 2004, Oct. 2005)**

R Long form of HTML is ______. **(March 13)**

(i) Hyper Text Mark up language

(ii) High Text Manipulation Language

(iii) Hyper Text Mainpulating Language.

(iv) High Text Markup Language

Ans. : (iii) Hyper Text Markup Language.

3. To display defination lists on your web page —— tag is used.

OR ______ tag is used to write the definition list. **(July 2018)**

(i) <DLIST> (ii) <OL> (iii) <LI> (iv) <DL>

Ans. : (iv) <DL>

4. Long form of HREF is —— **(March 2002)**

(i) Horizontal reference (ii) Hypertext reference

(iii) Hyperlink reference (iv) Hypermedia reference

Ans. : (ii) Hypertext reference

5. RGB code for BLACK color is ....

(i) FF0000 (ii) FFFFFF (iii) 00000F (iv) 000000

Ans. : (iv) 000000

6. Which of the following color name is not allowed to used in HTML ——

(i) OLIVE (ii) PURPLE (iii) ORANGE (iv) FUCHSIA

Ans. : (iii) ORANGE

7. —— tag is used to put a line break in HTML code. **(March 2003, 17)**
(i) <HR> (ii) <BR> (iii) <P> (iv) <TT>

Ans. : (ii) <BR>

8. To place the image into an HTML file —— attribute is used in IMG tag. **(March 2007, 11)**
(i) <URL> (ii) <ALT> (iii) <SRC> (iv) <HREF>

Ans. : (iii) <SRC>

9. VB Script can be executed in —— web browser.
(i) Netscape Navigator (ii) Internet Explorer
(iii) Both (iv) None of these

Ans. : (iii) Both

10. The long form of SGML is —— . **(March 2005)**
(a) Standard Global Machine Language
(b) Special Global Markup Language
(c) Symbolic Generalized Machine Language
(d) Standard Generalized Markup Language

Ans. : (d) Standard Generalized Markup Language

11. —— is the name of the web browser. **(Oct. 2006)**
(i) Embedded system (ii) Netscape Navigator
(iii) Oracle (iv) C++

Ans. : (ii) Netscape Navigator.

12. —— tag is used for scroll the text. **(Mar. 2009)**
(i) <STRIKE> (ii) <MARQUEE> (iii) HR (iv) None of these

Ans. : <MARQUEE>

13. COLSPAN attribute is used with _____ tag. **(Oct. 2007)**
i) <BODY> ii) <HTML> iii) <ITILE> iv) <TABLE>

Ans. : **(iv) <TABLE>**

14. <A> tag has attribute _______ which defines URL of the document to be linked. **(March 2008)**
i) SRC ii) HREF iii) VREF iv) REF

Ans. : **(ii) HREF**

15. In HTML, ______ attribute defines the name of the file in which the image is to be found. **(Oct. 2008)**
(i) ALIGN (ii) SIZE (iii) SRC (iv) BGCOLOR

ns. : **(iii) SRC**

6. ALIGN is not an attribute used with _____ <tag>. **(March 2010)**
(i) <BODY> (ii) <HR> (iii) <TR> (iv) <TABLE>

s. : **(i) <BODY>**

17. ______ is a tag in HTML. **(Oct. 2010)**

i) ALT ii) SCR iii) IMG iv) ALIGN

Ans. : **(iii) IMG**

18. ______ tag is used to put a horizontal rule in HTML Code. **(March 2012)**

(i) <HR> (ii) <BR> (iii) <P> (iv) <TID>

Ans. : **(i) <HR>**

19. For green Colour, RGB code is ______ **(Oct. 2012)**

(i) # FF0000 (ii) # 0000FF (iii) # 00FF00 (iv) # 00FFFF

Ans. : **(iii) # 00FF00**

20. ______ is not a tag in HTML. **(Oct. 2013)**

(i) IMG (ii) ALT (iii) BIG (iv) SMALL

Ans. : **(ii) ALT**

21. Border attribute is used in ______ tag. **(March 2014, 2016; July 2019)**

(i) <HTML> (ii) <P> (iii) <TABLE> (iv) <TITLE>

Ans. : **(iii) <TABLE>**

22. In HTML ______ is not a paired tag. **(Oct. 2014)**

(i) <B> (ii) <I> (iii) <BR> (iv) <TABLE>

Ans. : **(iii) <BR>**

23. The valid attribute of <A> is______. **(March 2015)**

(i) NAME (ii) SRC (iii) BGCOLOR (iv) HEIGHT

Ans. : **(i) NAME**

24. The ______ attribute of <OL> of HTML is use to change, Bullets of the List.

(i) START (ii) VALUE (iii) BULLETS (iv) TYPE **(Oct. 2015)**

Ans. : **(iv) TYPE**

25. The size of GIF format file is ______. **(July 2016)**

(i) Greater than BMP format file (ii) Less than BMP format file

(iii) Equal to BMP format file (iv) Greater than JPEG format file

Ans. : **(ii) Less than BMP format file**

26. The attribute BORDER in <TABLE> tag has the default value of ______. **(July 2017)**

(i) 2 (ii) 0 (iii) 1 (iv) None of these

Ans. : **(iii) 1**

27. In HTML, for red colour, RGB code is ______. **(March 2018)**

(i) #000000 (ii) #ff0000 (iii) #00ff00 (iv) #0000ff

Ans. : **(ii) #ff0000**

28. ______ tag is used to write the definition list. **(July 2018)**

(i) <UL> (ii) <DL> (iii) <OL> (iv) <DT>

Ans. : **(ii) <DL>**

29. Bulleted list in HTML is created by ________ tag. **(March 2019)**
(i) <UL> (ii) <OL> (iii) <B> (iv) <BR>

Ans. : (i) <UL>

30. ________ tag is used to create a row in table. **(March 2020)**
(i) <td> (ii) <th> (iii) <tr> (iv) <tt>

Ans. : (iii) <tr>

31. ________ tag is use for subscript in HTML code. **(Oct. 2011)**
(i) <sup> (ii) <subscript>
(iii) <sub> (iv) <super>

Ans. : (iii) <sub>

❑❑❑

Oct. 2003

Distribution of Marks- Questionwise and Topicwise

Sr. No.	Name of Topic	1 Mark Question		3 Mark Question		4 Mark Question		5 Mark Question		Total Marks
		Nos.	Total	Nos.	Total	Nos.	Total	Nos.	Total	
1	Operating Systems	1	1	3	9	3	12	-	-	22
2	Data Structures	1	1	3	9	1	4	-	-	14
3	C++ Programming	1	1	5	15	2	8	4	20	44
4	HTML	1	1	1	3	-	-	2	10	14
5	Total	4	4	12	36	6	24	6	30	94

Note : Answers of questions show in bracket of respectively given.

Instructions : 1) All questions are compulsory. 2) Figures to the right indicate full marks. 3) Use of any type of calculator is not allowed. 4) Draw neat diagrams, wherever necessary. 5) Header files and comments in C++ program are not necessary.

Q.1 A) Select the correct alternative and rewrite the following :

a) To place the image into an HTML file .... attribute is used in IMG tag.

(i) URL (ii) ALT (iii) SRC (iv) HREF (1)

b) The elements of a record are .....

(i) Homogenous (ii) Similar (iii) Non-homogenous (iv) Identical (1)

c) Windows NT is .... operating system.

(i) Multiuser (ii) Multitasking (iii) Multithreading (iv) All of the above (1)

d) The ability to take more than one form is called .... in object oriented programming. (1)

(i) inheritance (ii) encapsulation (iii) polymorphism (iv) data abstraction

Ans. : (a)(iii) SRC , (b) (iii) Non-homogeneous, (c) (iv) All of the above.,
(d) (iii) Polymorphism

(B) Answer any two of the following :

a) What is GUI ? State any four advantages of GUI. **(Ch.1 / Q-56 / Pg.-1-28)** (3)

b) Explain with flowcharts the following control structures : (3)

(i) Sequence logic (ii) Selection logic (iii) Iteration lagic **(Ch.2 / Q-10 / Pg.-2-7)**

c) What is HTML. ? State the advantages and disadvantages of HTML: **(Ch.4 / Q-1, 3, 4 / Pg.-4-1, 4-2)** (3)

Q.2 (A) Answer any two of the following :

a) State any six features of object oriented programming. **(Ch.3 / Q-40 / Pg.-3-29)** (3)

b) With a suitable example show a labelled diagram for link between two nodes having the information part and next pointer field. **(Ch.2 / Q-30 / Pg.-2-21)** (3)